

Reconfigurable automatic tasking system

Patent Number: EP0236744, A3, B1

Publication date: 1987-09-16

Inventor(s): KERR RANDAL HUGH; MESNARD ROBERT MARSHALL

Applicant(s): IBM (US)

Requested Patent: JP62214437

Application Number: EP19870101586 19870205

Priority Number(s): US19860838062 19860310

IPC Classification: G06F9/44

EC Classification: G06F9/46C4, G06F13/10E2

Equivalents: BR8701064, CA1267229, DE3787545D, DE3787545T

Cited Documents:

Abstract

A host system (14) is capable of executing interactively initiated operations or tasks. A device (10) is provided for executing a program having an interfacing program (12) loaded therein. The interfacing program (12) is adapted to define and to execute a task repeatedly. An interface is connected to the host system and to the device for executing a program that issues commands to and receives data from the host. An independent program is connected to the task that is defined by the interfacing program. This independent program adds generality to the interfacing program after the task has been defined.

(19) 日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開昭62-214437

(43) 公開日 昭和62年(1987)9月21日

(51) Int. C l. ⁵	識別記号	府内整理番号	F I	技術表示箇所
G 06 F 9/06	3 1 0 C			
G 06 F 9/46	3 4 0 B			
		G 06 F 9/06	3 1 0 C	
		G 06 F 9/46	3 4 0 B	

審査請求 有

(全33頁)

(21) 出願番号 特願昭62-27460

(22) 出願日 昭和62年(1987)2月10日

(31) 優先権主張番号 838062

(32) 優先日 1986年3月10日

(33) 優先権主張国 アメリカ合衆国 (U S)

(71) 出願人 000000709 インターナショナル ビジネス マシーンズ コーポレーション アメリカ合衆国 10504 ニューヨーク州 アーモンク
(72) 発明者 ランダル・ヒュー・カー アメリカ合衆国ニューヨーク州リツチフォード、アール・デイ1、ポツクス211番地
(72) 発明者 ロバート・マーシヤル・メスナード アメリカ合衆国ニューヨーク州エンディコット、ウイルマ・ストリート620番地
(74) 代理人 合田 潔 (外4名)

(54) 【発明の名称】再構成可能な自動タスキング・システム

(57) 【要約】本公報は電子出願前の出願データであるため要約のデータは記録されません。

【特許請求の範囲】

- (a) 対話的に始動された操作を実行することのできるホスト・システムと。
- (b) タスクの規定および実行を反復して行えるように適合されたインターフェーシング・プログラムがロードされた、プログラムを実行するための手段と、
- (c) 前記ホスト・システムと前記プログラムを実行するための手段とに接続され、前記ホス1～・システムに対してコマンドを発するとともに、前記ホス1～・システムからデータを受取るインターフェースと。
- (d) 前記インターフェーシング・プログラムによって規定されたタスクに接続され、前記タスクが規定された後に生じた前記ホス1～・システムの環境変化に前記タスクおよび前記インターフェーシング・プログラムを順応させる独立して実行可能な手段とからなる再構成可能な自動タスキング・システム。

【発明の詳細な説明】

A、産業上の利用分野

本発明は、タスクを生成、実行するためのシステムに関し、さらに詳しくは、動的に変化するホスト環境において使用するタスクを一般化するためのシステムに関する。

B、従来技術およびその問題点

人間資源およびコンピュータ・システムの分野では、毎日の作業に関連した反復的な作業に含まれる単調さを減らすことが求められている。例えば財務、科学または人事データを含むコンピュータ・レコードまたはファイルの更新、品質保証試験、編集操作等のタスクじこにおいて、繰り返される活動の量は膨大なものになる。このような単調な活動を人間が行うと、作業能率の低下、疲労、バーンアウト、そしてしばしば受取入れ難いなどのエラー率の増加を招く。

外部環境変化が予期されない、最も繰り返しの多い編集および試験の操作の際には、固定されたタスクつまりプログラム化されたルーチンが複数のファイルを更新するのに適当である。しかしながら、予期しなかった状況に応じて変化することを求める場合が、対話的な環境では数多くある。本発明が対象とするのは、後者のタイプの状況である。

ホス1～・システム環境の動的変化には、次のようなものが含まれる。

a、システム情報メッセージ(つまり、H 8 M)。

階層記憶管理プログラム)

b、エラー状態および関連したシステム応答(つまり、プログラム異常終了)

C 1 ささやかなものから破滅的なもの(つまり、

システム・クラッシュ)までのシステム障害従来、タスクは、変化しない環境状況、つまり環境の変化の仕方が予め予想されるいくつかのうちの1つに収まる状況に制限されていた。この結果、変化する環境に応答して様々

なタスクを単独で実行することのできるロボット～つまりプログラムの代りに、各々が単一のタスクだけを実行するロボットつまりプログラムが増えてきた。

変化を予測できない環境の性質ゆえに、そのような環境において適切に実行されるようにするにはいかにタスクを修正すべきかを事前に知ることができない、したがって、タスクを修正可能にするには、操作環境での予想されない状況を識別するとともに、あるモードで学習つまり獲得した経験を別のモードで実践する能力を備えなければならない。

本発明の目的は、人手を介さずに、反復してタスクを実行するシステムを提供することにある。

本発明の他の目的は、反復的なタスクを実行でき、かつ変化する環境条件に適合するシステムを提供することにある。

本発明のさらに他の目的は、対話を用いる環境における動的変化に順応するように、自動的に再構成することが可能なシステムを提供することにある。

C 6 問題点を解決するための手段

20 本発明によれば、再構成可能な自動タスキング・システムが提供される。ホスト・システムは、対話的“に開始された操作、つまりタスクを実行することができる。インターフェーシング・プログラムのロードされた。プログラム実行用装置が提供される。該インターフェーシング・プログラムは、タスクの規定と実行を反復するよう適合されている該ホスト・システムと該プログラム実行装置にはインターフェースが接続され、該ホストに対してコマンドを発するとともに、該ホス1～からデータを受取る。該インターフェーシング・プログラムは、タスクが規定された後、該インターフェーシング・プログラムに一般性を加える。

D、実施例

コンピュータ環境向けに意図された本システムは1次の3つの基本操作モードからなる。それは、教育(エデュケーション)、試験(テスティング)および再試験(リテスト)である。

これらの基本操作モードのそれぞれは、以下で述べるようなサブ・モードからなる。システムの要は5ESSION(セツション)プログラムである。これを用いることにより、セツションつまりタイム・ピリオドが前述の操作モードの1つに従って指定されたタスクの実行に費される。5ESSIONプログラムとセツション(タイム・ピリオド)のこのような違いを、本明細書では前者を英語のままで使うことによって表現している。

第1図には、本発明によるシステムの要素と。

要素間のデータの流れが示されている。

IBM(登録商標)社製のデータ通信アダプタ・カード付モデルPC/XTのようならコンソール。

すなわちいわゆるスマート・コンピュータ・ターミナル50 10が、コンピュータ・プログラムのロードと実行に適

用される。コンソール10は、英数字とグラフィックスを表示する。好ましくは解像度の高いディスプレイ、つまりカラーのブラウン管を含む。本実施例では2つのディスプレイ・スクリーンが用いられるが、これらは、一方はカラーで情報を表示でき、他方はモノクロームで情報を表示できるという事実によって区別される。本明細書では、ディスプレイをしばしばモニタまたはスクリーンと呼び、表示される情報をスクリーン・イメージと呼ぶ。5ESSIONと呼ばれるコンピュータ・プログラム12は、コンソール10に常駐する。5ESSIONプログラムの持つ機能は多く、そのすべては独立に用いることもできるし、他と関連させて使うこともできる。

コンソール10は、ホスト・コンピュータ14との通信、つまりホスト・コンピュータ14との間でデータのやりとりを行なうことができる。ホスト14の例は、IBMモデル3081プロセッサのようなメイン・フレーム・コンピュータである。

ホスト14、コンソール10、および下記の様々なファイルの間の通信は、5ESSIONソフトウェア12によって行われる。ホスト・コンピュータ14とコンソール10の間のそのような通信を可能にする機構がそこにはある。そのような機構の1つにエミュレータと普通呼ばれるものがあるが、これはIBM社からPC/VMバンド・システムとして入手できる。

図面で一括して参照番号15で示されているのは、ユーザの選択可能な入力ファイルであり、5ESSIONプログラム12に接続されている。

例えば、イメージ・ファイル16は、システムで下記のように使われるスクリーン・イメージを備える。別のファイル18はコマンドを記憶し、さらに別のファイル20は計時または性能データを記憶する。これら3個の入力ファイル15は同時に使用できる。記述を簡単明瞭にすめために、第1図ではイメージ・ファイル16、コマンド・ファイル18、および性能ファイル20をそれぞれ单一のファイルとして示しているが、条件が許すなら本発明のシステムではそれぞれ複数個のファイルで構成してもよい。実施例の記載は本発明の範囲を限定するものではない。必要に応じてファイルをどのように組み合わせて構成してもよい。

出力ファイルと呼ばれ、図面では一括して参照番号21で示される第2のファイルのセットは、5ESSIONプログラム12が生成したデータを受取るためのものである。差異ファイル22は、以前に記憶されたデータを最近生成されたデータの会話セットと比較して得られる情報を記憶する。この情報は、オペレータまたはユーザが下記のように再試験モードで再生機能を実行するときに最も有用であるが、教育モードでスクリーンを更新するのにも使うことができる。

出力性能データ・ファイル24は、システムと人間であ

るユーザの両方の、性能つまり効率の記録を備える。性能データ・ファイル24は、後でアクセスし、かつ、それに統いて解析することの可能な計時情報を備える。出力イメージ・ファイル26も5ESSIONプログラム12によってアクセス可能であり、コンソール10に表示されるスクリーン・イメージを表すデータを備える。

5ESSIONプログラム12に接続された出力コマンド・ファイル28は、セッションの間にオペレータが出10すコマンドを記憶する。

5ESSIONプログラム

5ESSIONプログラム12は、コマンドとスクリーン・イメージ全体の両方を捕えることができる。わかりやすくするために、本発明の主要な機能のいくつかをオーディオまたはビデオ・テープレコーダの特定の機能に例え、「レコード」「再生（プレイバック）」のような通常のレコーディング機能に関連した用語を使うことにする。

キャプチャリング（捕獲）またはレコーディング（つまり、PCにおける大容量記憶データ）が起こる時点から、5ESSIONそれ自身を含むいくつかのプログラムの1つを走らせることができる。これらのプログラムは、エミュレータ、エディタ、コンバータ、およびレビューである。

第1図に示されていない人間のユーザは、コンソール10を通してホスト14に常駐するシステム・プログラムをログオンする。統いて、ユーザはトゲリングによってコンソール10にアクセスし、コンソール10に前からロードされている5ESSIONプログラム12の実行30を開始する。

5ESSIONプログラム12は、ユーザをホスト14に再接続する。5ESSIONプログラムには、通常のキーボード（図示せず）または他の適当な手段によってPCコンソール10から呼び出すことができる。

5ESSIONという単語と適当なパラメータが入力される。コマンド5ESSIONについての構文は、次の通りである。

5ESSION <ファイルネーム> [、#*s傘p
*d傘n傘a v m s# p# d# c#ステップ
40 セイブオン/セイブオフ0/レジストラマスク=h] ここで、ブラケット [コでオプショナルであることが示されている前述のパラメータは、下記第1表の機能を表わしている。

第1表

使用するファイルのレベル。

*S スクリーン・イメージ・ファイルをセーブする。

*p 性能ファイルをセーブする。

*d 差異ファイルをセーブする。

50 傘n 同じファイルに重ね書きする。

* a 既存のファイルに追加する。
 VM VM (仮想記憶) システム・チェックングを含む。
 S# 、Sエクステンションを使う。
 P# 、Pエクステンションを使う。
 d# 、dエクステンションを使う。
 Q# 、Qエクステンションを使う。
 ステップ 各ユーザ・コマンドの再生と、それらの間でのボーズ。
 セイブオン/セイブオフ 第1のスクリーンからレコード・モードを入力する (または入力しない)。
 0/1 システム・スクリーンを1分離した白黒またはカラーのモニタに表示する。
 再生 再生モードで5 SESSIONを入力する。
 マスク=n ユーザ指定の取消し文字 (n)。差異ファイルを作成するときに用いる。
 システムに対するコマンドを生成するのを助けるのに最も有効なことは、特定コマンドを実行するために、特定のファンクション・キーをキーボード上に定義することである。5 SESSIONプログラム12においては、このようなファンクション・キーがデフォルトによって下記第2表のように定義されている。しかしながら、ユーザが通常のPCプロファイル・プラクティスに従って。コントロール・キーをどのように定義することも可能であることを理解しなければならない。以下のファンクション・キーの定義はデフォルト・システム定義であり、一般的なアプリケーションにとって最も有用であることがわかっている。

第2表

リーン・イメージを消す。
 CTRL-D ディレクトリ 指定されたブイレフーリ・マスク・ファイルをリストへする。
 CTRL-E エンド5 SESSION 5 SESSIONプログラムを終了させ、103へ戻る。
 CTRL-['] へ) Ld 5 SESSIONキーを定義するヘルプ・ウィンドウを表示する。
 CTRL-I 直接セーブ 現在のスクリーンをディスク上のスクリーン・ファイルへコピーする。
 CTRL-L ロック・キーボード パス・ワードに基づくキーボード・入力を禁止する。
 CTRL-M マクロ・ファイル マクロ (コマンド) ファイルメイク/エンド の開始と終了を切り替える。
 CTRL-N ノート・ツ・スクリーン 70文字のノートをスクリーンと共にセーブすることを可能にする。
 CTRL-P マクロ・ファイル コマンド・

6 ファイルの実行をプレイバック 始める。
 CTRL-Q クイットA プレイバック・ファイルの実行をやめて、キーボードをユーザに返す。
 CTRL-R レコード・トグル セイビング・オンまたはオフを切り替える。
 CTRL-S ステップ・モード 一度に1つのキーストロークでプレイバック・コマンドへ切り替える。
 10 CTRL-V VM 1～グル VM システムが使用中であることを明確にする。
 CTRL-W ウェイト 一時的にプレイバックを中止 (プレイバック) するように切り替え、ユーザ入力を許す。
 CTRL-X イグジット1-pc オペレーティング・システムへ出る。ユーザがEXITをタップすると、5 SESSIONへ戻る。
 20 ALT-D タイプ・ダウン カーソルが下がる。
 ALT-U タイプ・アップ カーソルが上がる。
 ALT-0 モノ・オンリー モノクローム・スクリーンだけを使う。
 ALT4 カラー・オンリー カラー・スクリーンだけを使う。
 ALT-2 モノ・アンド・カラー モノクロームとカラーの両方のスクリーンを使う。
 30 5 SESSIONプログラムの中からキーCtrl-F1を押してヘルプ・メニューを呼び出したときに表示されるスクリーンを下記第3表に示す。なお、各キーに対応するファンクションが英語で示されているが、その訳および意味は上記第2表の通りである。

第3表

EMULATOR FUNCTIONS

Enter	Caps-Lock	Clear
:	5 scroll-Lock	:
Ctrl-Break	78	MVS

40 5 SESSION FUNCTIONS Blank 5 screen : Ctrl-B Type Right : Alt-R Playback : Ctrl-P Directory : Ctrl-D Type Left : Alt-L Quit Play : Ctrl-Q End Session : Ctrl-E Type Up : Alt-u Record : Ctrl-R Instant Save : Ctrl-I Type Down : Alt-D Step Mode : Ctrl-5 Lmk Keys : Ctrl-L Memo On

y : Alt-0 VM Checks : Ctrl-VM
ake File : Ctrl-M Color
Only : Alt-I Wait
: Ctrl-V Note : Ctrl-N Mono & C
olor : Alt-2 Exit DO3 : Ctrl-X S
SESSIONプログラム12は、プログラム呼出し時に総括的な（ゼネラル）ファイルネームを指定することにより1周期的に（つまり毎日）走らせることができる。

このようなファイルネームは、セツションに関する重要な情報をセーブする必要ができたときにセーブ・スクリーンズ・オプションを指定した現在の日付であってよい。特定情報をセーブするために+ Ctrl -M（マイク・マクロ・ファイル）オプションが用いられ、該情報に関連してあるファイルネームが指定される。

この結果、ユーザがマクロ・ファイルをその作成直後にプレイバックできるようになる。

セツションのレコーディングがその名の下で行われるファイルネームが、呼出し時に指定される総括的な5ESSIONファイルネームであるならば、該5ESSIONファイルネームが変更されるまで、ファイルのプレイバックは不可能である。ファイルネームを変更するには、Ctrl-r1-Eで5ESSIONを終わらせ、かつ違うファイルネームで5ESSIONを再始動する必要がある。

なぜなら、呼出し時に生成される5ESSIONファイルは、出力用に開かれるファイルであり、入力と出力の両方として使うことはできないからである。したがって、その日その日のルーティン・オペレーションについては、現在の日付（例えば、AUG 27）のような属性的な（ジェネリック）SESSIONファイルネームを用いることが望ましい。

本発明の内容を理解するために、続いて実行例を説明する。以下の例は説明の目的だけでとりあげものであり、これによって本発明の範囲が制限されるわけではない。以下の例は、学生に151L言語のコマンドの使い方を教えるためのものである。この例では、"IBM Virtual Machine/System Product Editor Command and Macro Reference" (Endicott, NY. 5 second Edition 1982) に記述されているような、システム・エディティング・プログラム（例えばXEDIT）が必要とされる。

まず、ユーザは、次のPCコマンドを入力する。

5SESSION 151L*S

ユーザは、一旦レコーディングを開始したいポイントーを決めると、コンソール10からCTRL-Rを入力する。すると、システムはレコーディング・モードとする。*Sが元の5SESSIONコマンドで指定されたので、システムはキーストロークと、これから入力される

スクリーン・イメージを捕える。

CTRL-Rが入力された後、第2a図に示される最初のスクリーンは、新しく名付けられたファイル151Lに記録される。ISI LD EMO5CRIPTというファイルネームを持つVMデータ・セラ1～を生成するために、ユーザは次のコマンドを入力する。

XEDIT ISI LD EMO5CRIPT AI

このステートメントにより、生じつつあるセツション会話を忘れて、ユーザがXEDITシステム・エディティング・プログラムを実行できるようになる。したがって、ユーザが単にXEDITルーティンを実行し、XEDIT環境で操作するだけで、レコーディング・セツションがユーザにとって透過的に実行される。結果として生じて記録されるスクリーンが第2b図に示される。

この実施例では、著者が、学生に文字h1～h6によって識別されるサブ・ヘッディングまたはヘッダの使い方を教える教育パッケージを生成する。これらの文字および同様の文字は、タグとして知られている。時々このセ

20 ツションでは、第2a～第2P図に示されるように、与えられたスクリーンを凍結して獲得するために、著者がコンソールのアテンション・キーを押す。その後セツションを続けることができる。このように繰り返しアテンション・キーを使うことは、5ESSIONを使う著者にとって余分な仕事ではない。なぜなら、ホストへ・システムでの作業の際に、アテンション・キーを使ってシステム・レスポンスを得ることは普通に行われている。ENTER（入力）キーはアテンション・キー一例である。普通、アテンション・キーは、スクリーンの更新またはリフレッシュに使われる。

アテンション・キーの他の例は、CLEAR, PA1, PA2、およびファンクション・キーF1～F12である。

次に、著者はコマンドINPUTを入力して、XEDIT入力モードを呼び出す。結果として生じて記録されるスクリーンが第2c図に示されている。著作が次に入力するステートメントは。

:p, Paragraph CCR used like :p, This である。ここで、(CR) は

40 キャリッジ・リターン・キーを表わしている。文字:p、は、新しいパラグラフの始まりを示す。これらの文字に続くすべてのテキストは、パラグラフの一部として受け入れられる。結果として生じて記憶されるスクリーンが、第2d図に示されている。

著者が次に押すキーは、単にENTERだけである。この結果、著者が後での修正のために前のスクリーン・イメージをセーブできるようになる。

したがって、今は、第2d図に示されるスクリーン・イメージのコピーが2つ存在する。第2e図に示されるスクリーンを参照することにより、学生はテキスト上の1

5 I Lタグの効果を後で見ることができる。今度現われるスクリーンは、第2 f図に示される通りである。

続いて、著者は、

```
: h1. Head level one [CR
コニh2. t (ead level tw.
をタイプして入力する。
```

文字:h1. および:h2. は、テキストの中のヘッディングとサブヘッディングを示す。ヘッダ・レベルを示す数字は、1から6までである。結果として生じて記録されるスクリーンが第2 g図に示されている。

続いて、著者は、

```
: h3. Head level three [CH
3@>: h4. Head level four [C11]をタイプして入力する。結果として生じて記録されるスクリーンが第2 h図に示されている。
```

続いて、著者は、

```
: h5. Haad level five [CH3
: h6. +lead level sixをタイプして入力する。結果として生じて記録されるスクリーンが第2 i図に示されている。
```

著者がENTERキーを再び押して第2 i図の複製スクリーン・イメージをセーブすると、第2 j図に示されるように、後で該イメージを変えてテキスト上のI 5 I Lタグの効果を示すことが可能である。

今度のシステム・スクリーンは、第2 r図に示す通りになる。

リストを生成するために、著者は次のテキストを入力する。。

```
: s1. [CH3
: li. item1 [CH3
: li. item2 [:CH3
: es1.
```

文字:s1. は、単純(シンプル)リストの始まりを示す6文字:li. は、リストの項目つまりメンバを識別する。文字:es1. は、単純リストの終りを示す。結果として生じて記録されるスクリーンは、第2 l図に示されている。

テキストのI 5 I Lタグの効果を見るために、著者が再びENTERキーを押すと、複製スクリーン・イメージが生じるが、これは後で変更されて第2 m図に示されるスクリーンを生じさせる。

リスト(項目1と項目2)がレベル6のサブヘッディングの下に現れる。なぜなら、該レベルは、リストが開始する前に参照された最後のレベルだからである。今度現れるシステム・スクリーンは、

第2 n図に示す通りである。

続いて、著者は、

```
euL. [CH3
: li. item1 [CH3
: p. 5tuff for I [CH3: li
```

、1ten+2 [CH3
:euL。

をタイプして入力する。

文字:uL. は、無秩序リストの始まりを示し、文字:euL. は同リストの終りを示す。結果として生じて記録されるスクリーンは、第2 o図に示される通りである。

続いて著者がENTERを押すと、第2 p図に示されるようにX EDIT INPUTモードが終了する。

10 ユーザは、レコーディングを終えると、CTRL-Rを入力してレコーディング・セッションを終了する。しかしながら、XIE-DITシステム・プログラミの実行は依然続いている、望むならユーザーは普通はそこから出ることができる。また、レコーディング・モードが実現されていない間でも、ユーザーは依然としてSESSIONプログラムを実行している。SESSIONプログラムを終えるために、ユーザーはCTRL-Eをタイプする。・

SESSIONが実行中である間に、2つのファイルが20 作成された。一方はスクリーン・イメージ26(第1図)についてであり、他方はコマンド28についてである。

ここで、著者は以前に記録したあるスクリーン・イメージを修正するとともに、自動的に生成される基礎インストラクション・テキストにヘルプ・テキストを加え、ヘルプ・ファイルを生成したいと思うかもしれない。このようなことを行うために1次の構文でエディタが呼び出される。

EDITOR<ファイル名> [s1 h# c1
30 F 0/11上記簡略化したパラメータは、SESSIONコマンドに関して記述したもの(前記第1表)と同じ意味を持っている。

さて、著者は1次のコマンドを入力して実行例を進める。

EDITOR I 5 I L

エディタ機能では、ユーザーの定義可能な特定のキーが、デフォルトによって下記第4表のように割り当てられている。しかしながら、ユーザーは、通常のパーソナル・コンピュータ・プラクティスに従ってどのようにキーやキー・コンビネーションを定義できることを理解すべきである。

以下で述べるのは、エディタ・プログラムで使われるキーストロークのリストである。これらのキーは、すべてCfrl(コンソロール)キーまたはAlt(代替)キーと結びつけて使われる。すべての定義の形式は、Ctrl-<key> またはAlt-<key> であり、Ctrlと他のキー、またはAltと他のキーを同時に押す必要があることを示している。これらのキーは、エディタのヘルプ・パネルの中でも定義される(Ctrl-F1参照)。

第4表

ールドの型。
正／生成を許す。
ALT-C コピー・マーク 、マークされた領域をカーソル位置へコピーする。
CTIIL-BACKSPACE ラインの削除
ディスプレイから現在のラインを削除する。
CTRL-D ディレクトリ・指定されたブイレフ1-リ・リストティング ファイルを表示する。
CTRL-E エンド・エディタ エディタ・プログラムをF-3終了させ、00Sへ戻る。
CTRL-X イグジット～・エディタ PC オペレーティング・システムへ戻る。
CTRL-END スクリーンのトップ カーソルをスクリーンのトップへ動かす。
ALT-F4 ラインの終りまでの消去 カーソルからラインの終りまで文字を消す。
CTRL-G ゴーザ・モード エディタをゴーザ(GOTO)モードにする。
F1 ヘルプ エディタ・キーを定義するヘルプ・ウィンドウを表示する。
ALT-J ラインの接合 続くライン上のテキストを現在のラインの終りにつなぐ。
ALT-L マーク・ライン コピーまたはムーブ(移動)用のラインを付ける。
ALT-M ムーブ・マーク マークされた領域を現在のカーソル位置へシフトする。
'GDN 後続スクリーン ユーザを次のシステム/F8
ヘルプ・スクリーン・ディスプレイへ移す。
pcup 先行スクリーン ユーザを先行するシステムF7
ヘルプ・スクリーン・ディスプレイへ移す。
ESCセイブ・スクリーンズ ヘルプおよびシステム・スクリーンをセーブし、次のスクリーンを表示する。
CTRL-CRLF ・スプリット・ライン カーソルの所で現在のラインを分ける。
F9 スワップ・スクリーンズ 活動スクリーンをエディットに変える。
ALT-U アンマーク コピーまたはムーブのためにタグのついた領域のタグを取る。
ALT-R タイプ・ライト 普通のタイピングを指定する。

ALT-L タイプ・レフト キーがタップされた後でカーソルが左へ動くことを指定する。
ALT-D タイプ・ダウント カーソルが下がる。
ALT-U タイプ・アップ カーソルが上がる。
ALT-0 モノ・オンリー モノクローム・スクリーンだけを使う。
10 ALT-1 カラー・オンリー カラー・スクリーンだけを使う。
A) T-2 モノ・アンド・カラー モノクロームとカラーの両方のスクリーンを使う。
CTRL-110MIE スクリーンの底 カーソルをスクリーンの底へ移す。
F1 を入力して呼び出されるエディタ・ヘルプ・スクリーンは、第5表の通りである。なお、各キーに対応するファンクションが英語で表示されているが、その訳および意味は上記第4表の通りである。
20 第5表
Editor Help
-LINE CONTROLS--MARKING-Alt-F4 : Erase EOL Alt-C : Copy Mark Alt-J : Join Alt-L : Line Mark Ctrl-CRLF : Split Alt-M : Move Mark Ctrl-[3]ack space:Delete Line Alt-U:Unmark-5CrlEN CONTROLS--5PECI
30 AL COMMANDS-F1 : Help
Ctrl-B : Background
ColorSelect
F3 : Exit Without Saving
Ctrl-C : Command Entry F7
/PgUp : Show Previous Screen
Ctrl-D : Directory Listing
F8/PgDn : Show Next Screen
Ctrl-[E : End Editor F
40 9 : Swap Screens
Ctrl-G :
Go to a screen ESC: 5ve
5 screen and Continue Ctrl-N : Note Display on screen
Press any key to continue
エディタけ、SESSIONで生成されたファイルと用いられるスクリーン指向の編集プログラムである。エディタを、スクリーン・ファイル(、S#)、ヘルプ・ファイル(、H#)、そしてコマンド・ファイル(、C#
50

) にさえも用い、新しい情報の修正、生成の少なくとも一方を行なうことができる。

エディタは、システム・スクリーンをカラー・モニタに配置するとともに、ヘルプ・スクリーンをモノクローム・モニタに配置する。デフォルトの場合、W集可能な領域はヘルプ・スクリーンへ向かう。單一スクリーン・システムではヘルプ・スクリーンが表示され、トゲリングによってシステム・スクリーンがアクセスされる。

エディタは通常の編集機能を多く持っている。

領域をマークして移動、コピーしたり、ラインを挿入、削除することが可能である。これらの機能は、個々のヘルプまたはシステム・スクリーンの中だけでなく1両者の間でも実行することができる。

また、エディタは、カラーおよび属性の情報の定義を可能にする。背景と前景の両方の色彩を変更できるし、入力領域も同様に定義できる。

第3図には、ターミナル34に接続されたエディタ38が示されている。エディタ38に入力できるファイルは、以前に生成されたイメージ・ファイル26、コマンド・ファイル28、およびインストラクションつまりヘルプ・ファイル30である。

エディタ38は、新しいイメージ・ファイル26a、コマンド・ファイル28a、およびインストラクション・ファイル30aを生成することができる。エディタ38は、そのようなファイル2G a、28a、または30aの何れをも、それぞれ既存ファイル26、28、および30からか、またはそれらに関係なしに、生成することができる。

エディタは、スクリーンおよびコマンド・ファイルの存否に関係なく使用できる。5ESSION捕獲ファイルと結びつけて使うのが最も有効であるが、エディタは5ESSIONから独立して使うことができる。エディタによって生成または修正されたファイルを、再び5ESSIONに入力したり、またはエンジケータに入力したりできる。

ヒューマン・ファクターのテストにおいて、エディタはプロトタイピングのツールである。システム全体を、普通のテキスト・エディタのスピードで模倣することができる。これらのスクリーン・は、視覚的プロトタイプとして使えるだけでなく、エジケータ・プログラムがアクセスするときの基礎的なプロトタイプとしても使える。

エディタは、エラーをドキュメント～(文書化)するのに使える。捕えた問題が大きすぎてどのスクリーンにもノートの形で記述できない場合、エディタを用いて適用可能なシステム・スクリーンについて全部のスクリーン・テキスト～を生成してもよい。

エディタを用いて、テスト・ケース(コマンド・ファイル)の修正もできる。もつとも、この目的のためには、コンバート(CONVERT)が下記の“ように設計

されている。

どのエディタの中でもスクリーンとヘルプを変換して使えるけれども、エディタ・プログラムを用いると、これらのスクリーンを生成するとともに、それらが提示されるやり方でそれらをながめることができる。エディタは、コマンド・ファイルとともに、ヘルプおよびスクリーン・ファイルの両方を修正することができる。これが5ESSION捕獲教育パッケージにヘルプを付加する最も簡単な方法である。

10 エディタに特有のことではないが、エミュレーション・ツールはその用い方を通して説明することができる。マクロ・ファイルを使ってスクリーン・ファイルを生成することにより、操作方法の全体の文書化、および教育の基礎の形成が可能になる。マクロにおいて教育の目的で急速に発生しそうな複雑なタスクを5ESSIONの中で捕えることができる(ブレイバッケンS)とともに、記述テキストを付加することができ(エディタ)、ユーザ・ベースでマクロを実行することができる(エジケータにおけるユーザ・モード)。

20 、さて、著者はCTRL-Gを押してGOTOモードを呼び出す。実行例で既に記憶済のものの中から最初に修正されるスクリーン・イメージは。

イメージ・ファイルのページ5にある(第2e図)。著者は、CTRL-05をタイプすることにより、第2d図の複製イメージをアクセスする。次いで、普通の編集機能を用いて該スクリーンを修正し。

第2e図に示されるイメージに到達することができる。ここでユーザは、ESC(エスケープ)キーを押すことにより、第2d図のこの修正されたバージョンをセーブする。

30 第4a～第4p図には、第2a～第p図に示されるイメージにそれぞれ関連・対応するテキストを含むヘルプ・スクリーンが示されている。第4a～第4n図は、編集セッションの際に入力された情報テキストを表す。第4o、4p図は、エディタおよびエジケータ・プログラムによって自動的に生成される基礎インストラクション・テキストを表す。

ここで、第4a～第4P図のスクリーンに記された英文の大意を記す。

40 まず、各スクリーンの先頭に共通して表示されているKeystrokes entered to get here:の次の行には、各スクリーンの1つ前の段階のスクリーンで入力されたキーストロークが示されている。

また、各スクリーンの末尾において、Keystrokes needed for next screen:の次の行、またはtype。

pressの指示の直後には、各スクリーンの次の段階のスクリーンへ進むために入力すべきキーストロークが示されている。したがって1例えば第4a図から第4b

図に移るためには、`xedit isilden + o`
`5cript a1`とENTERを入力しなければならない。

第4a図は、「A」ディスクにI S I LD E M O
 5 C R I P Tというデータ・セットを生成するために必要なコマンドの入力を求めている。第4b図は、標準的なX E D I Tスクリーン(第2b図)が表示されている状態で、入力モードを入力することを求めている。第4c図は、:p、コマンドが新しいバラグラフの始まりを示すことを述べるとともに、:p、コマンドを実際にタイプすることを求めている。第4d図は、I 5 I L がこのタグをどう扱うかを見てみよう述べている。

第4e図は、X E D I Tで使うテキスト・ファイルのフォーマツ]・が書かれたI 5 I Lファイルのフォーマツーとほとんど関係がないこと、:p。タグによってテキスト内での分離とライン・スキップが生じることを述べ、最後にはエディタの再入力を求めている。第4f図～第4h図は、:h 1 、 1 l a a d l e v e l o n e (ヘッド・レベル1)～:h G。

Head level six (ヘッド・レベル6)の入力を求めている。第4i図は書かれたI 5 I L がどのように見えるか見ようと述べている。第4j図は、オンラインではわかりにくいけれども、レベル1のヘッドによって新しいページが該ヘッダをトップにして印刷されること、レベル2のヘッドはボールド体のダブルストライクで印刷されること、レベル3のヘッドはイタリック体のダブルス1～ラインで印刷されること、レベル4のヘッドはボールド体で印刷されること、レベル5はイタリック体で、かつテキストとインラインで印刷されること、レベル6のヘッドはイタリック体、そしてテキスト・サイズで、かつテキストとインラインで印刷されることを述べている。第4i図は、I 5 I Lのリストには単純、無秩序、秩序の3種類があることを述べ、単純リストを見てみよう呼びかけている。第4i図は単純リストの説明(第21図に関して既に述べた)である。

第4m図の説明も第2m図に関して既に述べたものと同じである。第4n図は、違うタイプのリストを見てみようという呼びかけである。

スクリーン・イメージの修正バージョンをセーブした後、システムは次のスクリーン・イメージに進む。ここで著者は、第21図で示されるように元々セーブされていたイメージの修正を望むかもしれない。これを行うには、C T R L - G (G o T o) 1 0を押す。すると、第21図のスクリーンを適当に修正することが可能になる。E S Cキーを押すことにより、修正されたイメージは第2j図に示されるようにセーブされる。

同様に、ページ13に現れる第21図の複製を修正して、第2m図に示されるようなスクリーン・イメージを作ることができる。

これで教育パッケージは完成した。該パッケージはフロッピー・ディスクにロードされたり、またはネットワーク能力を使って教育がなされる任意の場所に送られたりする。一旦受取られた教育パッケージを、場所毎のバリエーションをもたらすエディタ・プログラムを使って修正することができる。

孜J I iニエー

教育モードにおいて、システムは、自動著作ツールである。教育の著者は、5 E S S I O Nプログラム実行時にホストを使っているオペレータである。記録されたスクリーンおよびコマンドのどれもが教育文脈で有用となるように記憶される。

5 E S S I O Nプログラムが実行を終える否や教育モードを入力することにより、捕獲ホスト・セツションがインストラクション個人教授形式においてP Cレベルでシミュレートされる。

第5図には、教育ドライバ・システムが示されている。イメージ・ファイル26とコマンド・ファイル28は、第1図に関連して既述した。I 5 I Lコマンドを教えるのに用いたセツションの実行例において、スクリーンつまりイメージ・ファイル26は、レコーディング・モード時のホスト・セツションの際に生成されたスクリーン・イメージのすべてのコピーを結局備える。これらのスクリーン・イメージは、単一のイメージ・ファイル26に引き続いて記憶される。同様に、コマンド・ファイル28も実行例のセツションの際に生成される。インストラクションつまりヘルプ・ファイル30も、教育ドライバ・システムで付隨的に備えられている。

教育ドライバ・システムの要は、通常の多位置モード・スイッチ33が常駐する教育ドライバ32である。ドライバ32がその一部をなすコンピュータ・ターミナル34は図示しないが標準的なコンソールであり、ユーザがプログラム・ドライバ32と通信するためのキーボード等の久方手段を含む。図面で一括して参照番号36で示されているのはモードであり、モード・スイッチ33によってドライバ32に接続されている。モード・スイッチ33を適当にセットすることにより、各モードを一意的にアクセスすることができる。本実施例では、8個のモードを使うことができる。

モード36a～36cは個人教授アプリケーションで、モード36d～36fは参照アプリケーションでそれぞれ用いられる。モード36g、36hはニーテリティ/ドキュメンテーション・モードである。

5 E S S I O Nログ、5ム12(第1図)により以前に生成されたものと同様の性能データ・ファイル24もドライバ32に接続されて、必要に応じ計時データを受け取る。計時データは、ある事象とそれに対するキーボードを通じての人間の反応との間の時間間隔を表わす。この時間間隔が大きくなるほど、もちろん人間オペレータは無能である。

50

教育ドライバ32は、コマンド・ファイル28、イメージ・ファイル26、および存在すればヘルプ・ファイル30を連係し、これらのファイルを個人教授風に提示するエジュケータ・プログラムを含む。

エジュケータ・プログラムは、個人教授と訓練、オンライン・ヘルプ、提示および表示とエラー文書化をサポートする操作モードを持つ。

エジュケータ・プログラムは、著者すなわち教師のために教育パッケージを著作する会話形教育ツールである。

SESSIONプログラム12(第1図)を使ってホスト・システム14を捕えることにより、エジュケータ・プログラムは、該ホスト・システム14をシミュレートでき、さらに著作する必要なしで基礎インストラクション・テキストへの提供さえも行うことができる。完全な教育パッケージを生成するために、エディタ・プログラムは、上述のようにスクリーン1つずつのヘルプ・テキストを生成しなければならない。前述の実行例において、そのようなスクリーン1つずつのヘルプ・テキストは、エディタ・セツションの際にシステム・スクリーンの一部のために生成される。残りのシステム・スクリーンは、教育ドライバ32が生成する基礎インストラクション・テキストを持つ。

エジュケータ・プログラムを呼び出す構文を次に示す。
EDUCATOR<ファイルネーム> [、# s# c# h#モードO/1] ここで、ブラケット[]でオプショナルであることが示されている前述のパラメータは、下記第6表の機能で表わしている。

第6表

- 、# 使用するファイルのレベル。
- S# スクリーン・レベル・ファイルを使う。
- C# コマンド・レベル・ファイルを使う。
- h# ヘルプ・レベル・ファイルを使う。
- モード 開始モード。ユーザ、ステップ、オート、コマンド。
- リスト、ゴーツの1つ。
- 0/1 出力をモノ/カラー・ディスプレイへ強制的に向ける。

エジュケータ・コマンドで1. #が指定されていなければ、システムは未指定ファイルのそれぞれについて最も高いレベルを設定する。パラメータ・レベル(例えば4)は、使用中のファイルのデフォルト・レベルを指定されたレベル(例えば、第4レベル。s 4, c 4, h 4)に設定する。使用されないときのデフォルト・レベルは、そのファイルのタイブーコマンド(、C)、スクリーン(、S)、またはヘルプ(、H)の存在レベルの最高値となる。指定レベルは、1から99の間の数であつてよい。

個々のパラメータはファイルネームから分離されていなければならず、かつ、互いに少なくとも1スペース離れてなければならない。例を次にあげる。

EDUCATOR151L 5TEP 1上記エジユケータ・コマンドの例によれば、ユーザが実行例に関連してエジュケータ・プログラムを呼び出すことが可能になる。

カラー・オンリー(つまり、出力デバイス1)は、(例えば、2スクリーンPCで)カラー・スクリーンが使用可能ならば、エジュケータ・プログラムの出力をカラー・スクリーンへ向ける。2スクリーン・システムが使用されているならば、システム・スクリーンをディスプレイの1つに配置することが時には有用である。

モノクローム・オンリー(つまり、出力デバイス0)は、エンジュケータ・プログラムの出力をモノクローム・スクリーンへ向ける。

コマンド・ファイル(つまり、C#)は、コマンド・ファイルだけについてファイル・レベルの指定を許すレベル・パラメータのバリエーションである。最近のバージョンのコマンド・ファイル(バリュー・エクステンションが最高のファイル)を使う代りに、指定、値を持つファイルが使われる。

20 指定ファイルが存在しないならば、存在レベルが最高のファイルが使われる。

ヘルプ・ファイル(つまり、H#)も、レベル・パラメータのバリエーションであり、ヘルプ・ファイルだけについてファイル・レベルの指定を許す。

スクリーン・ファイル(つまり、S#)もまた、レベル・パラメータのバリエーションであり、スクリーン・ファイルだけについてのファイル・レベルの指定を許す。オート(AUTO)モードが2スクリーン・システムで指定されると、システムは7秒間休止し。

30 コマンドをタイプし、3秒間休止し、このような循環をしながら次のスクリーンへ至る。単一スクリーンでは、オート・モードは2秒間休止し、コマンドがタイプされた状態でシステム・スクリーンを3秒間表示し、このような循環をしながら次のスクリーンへ至る。ファイルの終りに達すると、

該サイクルが再び第1のスクリーンで始まる。

コマンド(CMND)モードが指定されると、

40 システムはユーザがコマンドをタイプするのを待ち、統いて正確にマッチするものを求めてリスト・モードとともに使用可能なコマンドのリストを探索する。マッチするものが見つからなければ、入力されたストリングを備えるすべてのコマンドがマークされる。マークされたコマンドが1つだけならば、そのスクリーンがアクセスされる。1以上のコマンドがマニツされたなら、マークされたコマンドのウィンドウがユーザ・オプションとして表示される。マークされたコマンドがなければ、COMMAND NOT FOUND(コマンドが見つからない)メッセージが表示される。

ゴージ(GOTO)モードが指定されたなら、システム50はユーザがスクリーン番号を入力できるウィンドウを表

示する。エジュケータ・プログラムは、指定されたスクリーン番号に分岐する。番号が教育パッケージのサイズよりも大きいならば、最後のスクリーンがアクセスされる。対応するヘルプ・スクリーンは、適当なゴージ指定スクリーンに連係される。

リスト(L I S T)モードが指定されると、システムは教育パッケージ内のすべての使用可能なコマンドのウィンドウを表示する。エントリ・タイプのキーは、コマンドとして表示されない。コマンドは、関連するスクリーン番号とともに表示される。ウィンドウの中でリストをスクロールしてコマンドを選択することが可能である。選択されたコマンドは、教育パッケージの中で発生する順番で表示される。スクリーン1つについてコマンドを1つだけ選択できる。スクリーンが表示されると、そこにあるすべてのコマンドも表示される。したがって、スクリーン1つにつき最初のコマンドを選ぶことが許される。その場合、他のコマンドは抑止される。

ステップ(STEP)モードが指定されると、システムはデータ・キーのどれかが押されるのを待つ。データ・キーを押すとすぐに、全コマンドがスクリーンにタイプされる。次に、ステップ・モードは別のデータ・キーが押されるのを待つ。

第2のデータ・キーが押されると、すぐに次のスクリーンがアクセスされる。

ユーザ(USER)モードが指定された場合。またはどのモードも指定されなかった場合、システムは、ユーザが正しいコマンドを入力して正しいコマンド・エントリ・キーを押すのを待つ。正しいコマンドを1つ入力するのに、3回のチャンスが与えられる。第6a～第6d図を参照して説明すると、第1のエラーに基づいて、第6a図に示されるようにメッセージが表示される。第2のエラーに基づいて、システムは下正確な文字に対する訂正を行う、ユーザは、第6b図に示される指示された訂正(つまり、無関係の文字の消去)を終えなければならない。第3のエラーでは、第6c図に示されるように適当なエントリがシステムによって表示されるとともに、次のスクリーンがアクセスされる。ユーザがコマンドに対するすべての必要な訂正をなした後に下正確なアテンション・キーを入力すると、付加的なエラー・メッセージが1例えば6a図に示されるように表示される。

エジュケータ・プログラムで使用可能なキーストロークが下記第7表に示されている。これらのキーの定義および機能は任意であって、設計者の要求または好みに応じて変化できることを理解すべきである。本実施例では、IBMパーソナルコンピュータ(p c)に使われているキーボードが下で説明されている。これらのキーはすべてコントロール(Ctrl)または代替(Alt)キーと結びつけて使われる。すべての定義の形式は、Ctrl-<key>またはAlt-<key>

>であり、ユーザがCtrlと他のキー、またはAltと他のキーを同時に押す必要があることを示している。

第7表

ードにする。

イルの始まりに置く。

CTRL-Cコマンド・モード エジュケータをコマンド・モードにする。

CTRL-E エンド・エジュケータ エンジュケータ・プログラムを終了させ、DO3へ戻る。

10 適切なファイルのすべてが
閉じられる。
CTRL-END ファイルの終端 ユーザを
教育パッケージの終端へ移す。
CTRL-G ゴーツ・モード エジュケータをゴーツ・モードにする。ユーザがスクリーン番号を入力するライ
ンドウを表示する。

CTRL-F1 ベルブ エジュケータ・キーを定義するヘルプ・ウィンドウを
20 表示する。どのキーもホスト・コントロールへ戻る。

CTRL-F9 ヘルプ・スクリーン システム
・スクリーンからヘルプおよびインストラ
クション・スクリーンへユー
ザを交換する。

CTRL-PGDN 後続スクリーン ユーザ
を次のシステム/ヘルプ・スクリーン・ディス
プレイへ前進させる。

CTRL-N ノート・ツ・スクリ、ーシステム
・スクリーンと関連・トグル 連するノート
・エリアを該スクリーンの底に表示する。

CTRL-PGUP 先行スクリーン ユーザ
を先行するシステム/ヘルプ・スクリーン・デ
イスプレイへ戻す。

CTRL-Q 照会モード 次のノート
を求めてスクリーン-5ステップ・モード エ
ジュケータをステップ・モードにする。

CTRL-U ユーザ・モード エジュケ
ータをユーザ・モードにする。

40 CTRL-III ライト・ア・ノー1～70文
字のノートを書いてスクリーンと共にセーブする
ことを可能にする。

スクリーンの底の近くにウ
インドウを表示する。ENTE
Rキーを押すと、直ちにノ
ートはスクリーンと共にセ
ーブされる。

CTRL-X イグジツイ-PCオペレーテ
ィング・システムへ戻る。ユーザがEXIT
をタイプするとエジュケ

タへ戻る。

ALT-R タイプ・ライ1～ 普通のタイピングを指定する。

ALT-L タイプ・レフト キーがタイプされた後でカーソルが左へ動くことを指定する。

ALT-D タイプ・ダウソ カーソルが下がる。

ALT-U タイプ・アップ カーソルが上がる。

ALT-0 モノ・オンリー モノクローム・スクリーンだけを使う。

ALT-1 カラー・オンリー カラー・スクリーンだけを使う。

ALT-2 モノ・アンド・カラー モノクロームとカラーの両方のスクリーンを使う。

Ctrl-F1 を入力すると現れるエジュケータ・ヘルプ・スクリーンは、下記第8表に示す通りである。なお、各キーに対応するファンクションが英語で示されているが、その訳および意味は上記第7表の通りである。

第8表

Ctrl-U USIERmode -Interactive learning (、tri-55T EP mode -sai+i-automatic sequence of images: any key to move on Ctrl-A AUTMode' -auto+aaatic sequence of images with tia+e delay

Ctrl-G GOTOMode -jump directly to specifiednumt迫る' screen

Ctrl-CCMND u+ode -co+wand mode、enter any string of characters command file is searched for specific command-ENTERkey begins searchCtrl-L LIST mode -select commands from a list of all that exist -ENTERkey begins displaying those selected

Ctrl-N NOTE 11 mode -read any existing note about the system 5creen Ctrl-fil WRITE mode-write note commenting about a-ENTERkey writes note to diskCtrl-F9 5uAP 5crn -single 5creen systems on

ly; displays help 5creenCtrl-F7/Ctrl-PgUp = Backward Ctrl-F8/Ctrl-PHDn = ForwardCtrl-HOME = First 5creen Ctrl-END = Last 5creenCtrl-E/Ctrl-F3 = End ProgramAlt-0=Mono Only Alt-1=Color Only

10 Alt-2=Both 5creensPress any key to return to system 5creen 教育ソフトウェア・パッケージを教育ドライバ32によって実行するために、ユーザはコマンド・エジュケータとファイルネームを入力する。ファイルネームは、以前に生成された教育パッケージと同じ名nfでなければならない。実実行例でのファイルネームは15ILである。

本実施例では、本発明を実施するのに2つのスクリーンを用いている。第1のスクリーンはスクリーン・イメージを順次に表示し、第2のスクリーンはスクリーン・イメージと並行して呼び出されるコマンドを表示する。しかしながら、他の実施例では、0/1出力デバイスのパラメータを使って、スクリーンとコマンドの両方を同じターミナルつまりコンソールに表示することができる。教育ドライバには2つの基本モードがある。個人教授・訓練モードと、オンライン・ヘルプまたは参照モードである。2スクリーン・システムでは、システム・スクリーンがカラー・モニタに現れ、ヘルプ・ファイルがモノクロームに現れる。

30 インストラクション・テキストつまりヘルプ・スクリーンは、システム・スクリーンに関連して編集された情報を表示する。システム・スクリーンに入力する必要のあるコマンドも表示される。

実行例において、最初のコマンドはXEDITISI DEMO5CRIPT A1である。

これがユーザの入力したコマンドである。タイプされた文字は、実際のホス]～・システム上のようにシステム・スクリーン上に現れる。ユーザが該スクリーンに必要とされるコマンドをタイプし、かつ正しいエントリ・キー(ENTER, PFキー、またはCLEAR)を押すと、続くシステム・スクリーンと共に該スクリーン用のインストラクション・テキストも表示される。該システムは工SIL教育パッケージにおいて捕獲されてしまっている。

上述のように、テキスト画面4a～4pは、それぞれスクリーン・テキスト画面2a～2pに関連している。ユーザはこの機能での操作を続け。

必要なインストラクションをタイプする。インストラクションを入力するのに必要なキーストロークは必ずしもコマンドに特有でなくてもよいが。

50 コマンド自身は正しくなければならない。

また、上述のように、不正確なコマンドがスクリーンに入力された場合、ユーザには複数のレベルが与えられる。第6a～第6d図は、与えられた3つのエラー・レベルを示す。第1のエラー・レベルでは、メッセージ（第6a図）が、入力の際にエラーが生じたことを示す。同じコマンドについて第2のエラーが生じると、足りない、または不正確な文字のどれもがシステムによって訂正され、かつ1色で強調される（第6b図）。応答に必要とされない文字は、どれも別の色で強調される。ユーザは、関係のない文字の消去を求められる。第3の不正確な試行では、システムによって正確な応答がスクリーンにタイプされ（第6c図）、スクリーンの底に短いメッセージが表示されるが、そこでは正しい答が表示中であることを示す。なお、以上の説明は、そのまま第6a～6d図の英文の大意である。不正確なエントリ・キーを押した場合（例えば、ENTERが求められているのに、PF5を押した場合）、適切なメッセージがスクリーンの底に現れ（第6d図）、ユーザに対してENTERの使用を指示する。ユーザは、ソフトウェア・パッケージを通してファイルの終端まで作業を続行し、インストラクションおよびヘルプ・テキストの要求するコマンドをリイブして、シミュレートされたシステムにおいてシステム・スクリーンを進んで行く。

上述の実行例では、エディタ・プログラムを使って生成されたヘルプ／インストラクション・テキストが第4n図に示されるスクリーンで終了する。該スクリーン（第4n図）で始めると、このスクリーン・イメージがエミュレータ・プログラムによって生成され、基礎インストラクション・テキストと呼ばれる。

オンライン・ヘルプ・モードは、リスト・モード機能を用いると最もよく機能する。この機能は、

システム・スクリーンにウィンドウを表示する。

ウィンドウには、この教育パッケージの中で使用可能なすべてのコマンドのリストが表示される。

コマンドとコマンドが現れるページ番号の両方が表示される。

該リストは、完全なスクロールおよび選択が可能である。ユーザは、リストをスキャンし、nだけと思う任意の項目の番号の前にS（「セレクト」を意味する）をタイプし、ENTERキーを押すことができる。ENTERキーを押すと、システム・スクリーンおよびヘルプ／インストラクション・スクリーンが最初に選択された項目に関連する。ユーザが任意のキーを押して自動的にステップ・モードを入力すると、システム・スクリーンにタイプされたコマンドを覗、続く任意のキーを押して結果として生じるスクリーンを見ることができる。対応するヘルプ／インストラクション・テキスト・スクリーンが同時に表示される。リストから1つ以上の項目が選択された場合は、再びどれかのキーを押すと、次に選択されたスクリーンがステップ・モードへ移る。ユーザがス

テップ・モードにとどまってリストから選ばれた全項目が表示されるまで、このようなことが続く。

この過程の間のいつでも、ユーザは、リスト・モードを再入力して、他のスクリーンを選んだり、以前選択したスクリーンの選択を取り消したり、既に見たスクリーンを見たりすることができる。

このような以前に表示されたスクリーンは、アスタリスクを付けて注釈される。このような種類のオンライン・ヘルプはタスク志向と呼ばれる。そして、ユーザに対して、コマンドを使うスクリーンのタイプ、コマンドの構文、そしてそのようなコマンドを入力して得られそうな結果を示すのに使われる。

ードライバのオンライン・ヘルプ・モード従来のオンライン・ヘルプのはとんどの形式はメニュー駆動式であり、複雑かつあいまいである。

ユーザは、ヘルプを探索して得た項目の使い方に關して完全な理解を得ない。構文がしばしば示されるが、実際の使い方はほとんど説示されない。

しかしながら、エミュレータ・プログラムは、タスク志向式でオンライン・ヘルプを提供する。これは、指定された項目について、構文と文脈上の使用法を示すことによって達成される。

オンライン・ヘルプ能力を利用する場合、本発明によって設計された#汀パッケージはより有効なものになる。このようにして、教育パッケージは教育以上のものになる。ユーザがシステムの専門家であっても、依然有用である。常時、該パッケージをPCレベルでロードすることができるとともに、トゲリングによって、ホスト・システムの問題について参照するために該パッケージをアクセスすることもできる6項目特有ヘルプ・パッケージを、例えば、メニュー駆動式システムのすべてのオプション、あるいは、ワード・プロセッサのタグのすべての動作を説明するように設計することができる。リスト項目としてあげることのできるものは何でも、タスク志向オンライン・ヘルプのサブジェクトになれる。

エミュレータの中では、リストおよびコマンド・モードが、オンライン・ヘルプについて最も有用である。リスト・モードは、パッケージの中の使用可能なコマンドすべてのウィンドウを表示する。

これは、スクロール可能かつ選択可能なリストであり、ユーザがヘルプを望む領域について選択をなすことを可能にする。

コマンド・モードは、リストが示されない点を除くと、リスト・モードと同じである。リストが示されない代りに、ユーザが情報を望む文字列をタイプすると、リストが探索される。コマンド・モードで実行される探索では、まずユーザ・エントリとリスト上のコマンドとの間の完全な突合せ（マツチ）の配置が試みられ、次にサブストーリング・マツチの配置が試みられる。効果は、リスト・モードの場合と同じである。ユーザには、要求し

50

たものについてのヘルプが示される。リストの中のコマンドは、入力タイプ・キーによって分離される。すなわち、選択可能なコマンドのそれぞれの前にはエントリ・キーが先行し、同じく後にはエンターリ・キーが続く。多くの項目がホスト・システム・スクリーンに現れるかもしれないが、リスト・モードの項目の1つだけが選択可能である。

これは、スクリーンから項目を1つ選ぶと、他のすべての項目も表示されるという理由による。

教育モードの生成には1代替例もある。それによると、以下のステップが含まれる。

a) 上述のように、5ESS工ONがスクリーンとコマンドを捕獲するのに代って、エディタを用いてこれらを生成し。

b) 下記のようにコンバートを用いてテキスト・ファイルからコマンド・ファイルを生成するとともに5ESS1ONを用いて該コマンドをプレイバックし、関連するイメージ・シーケンスを自動的に捕え、

C) 任意のエディタを用いてイメージおよびGML様タグのセットを使うコマンドのテキスト・ファイルを生成し、コンバートを用いてエミュレータ・プログラムが使うスクリーンおよびコマンドのファイルを生成する。

賦1し1ニ下

・試験は、繰り返しの、または同様のコマンド・セットを使う場合でも、実行の度に違うデータを出すと期待される項目に関連する。そのような試験の2つのカテゴリは、ヒューマン・ファクター試験および性能試験である。ヒューマン・ファクター試験は、基本または拡張のどちらかに分類することができる。

基 ヒューマン・ファクター

基本ヒューマン・ファクター試験は、プロトタイピングの理論に関する本発明のシステムによれば、視覚的に意味のあるスクリーンとコマンドの両方の高速プロトタイピングが可能になるとともに1機能を示すことも可能である。プロトタイプが完成すると、直ちに任意の数の人間に送ることができる。そして、該プロジェクトについて膨大なコーディング時間を費す前に、これらの人間からコメントを集めることができる。プロトタイプのレビュー・承認が終るとすぐに、後でコンバート・プログラムに関して説明するように、既にPCレベルで設計されたスクリーンを適切な言語に変換してホストにアップロードすることができ、その結果、作業を始めからホスト・システム上に置換する必要がなくなる。スクリーン・デザイン自身も、このようにしてより一部シンプルにできる。完全なスクリーン・エディタによって、はどんなシステム・ツールと他のプロトタイプのデザイン時間を要する色彩定義およびフィールド定義文字をバイパスして、プロトタイプの生成はなされる。

Pヒューマン・ファクター

ラボラトリ環境での試験、計時データの収集。およびユーザが行って見るもの理解は、すべて拡張ヒューマン・ファクター試験の一部である。

これは、製品がどの程度後に立つかを決定するのに実行される試験である。本発明のシステムは、

次のような能力を持つプログラムを含む。

a) ユーザが1つのスクリーンから別のスクリーンへ進むのに要する時間（システム・ディレイを含む場合と含まない場合がある）を測定する。

10 b) 各スクリーンでユーザが入力したキーストロークを正確に示す。

C) いつでも他人が見たリビューしたりできるようにユーザの経験全体を再生成する。

この結果、トップウォッチ、オーディオ・テープ・レコーダ、ビデオ・カメラのような、従来のヒューマン・ファクター測定・am器具が、かってのように置換できなかった状態を脱した。今ではこれらを全部取り除いてもよい。さらに、ラボラトリでもはや試験を行う必要がない。試験はサブジェクト自身の卵焼で行える。これは

20 20、各ユーザがPCつまり同様のスマート・ターミナルを持ちさえすれば、従来よりもはるかに多数の視聴者を相手にできることを意味する。

性能チャートは、第9表のようなフォーマットを持つ。

第9表

経過時間 中間の時間 タイムスタンプ

command : <コマンドのテキスト>) s

end :

(back :

command : <コマンドのテキスト>) sen

d :

(back :

縦の第1列は、時間0 : 00 : 00、00に始動し、コマンド毎に増加する。第2列は、コマンド毎に時間0 : 00 : 00、00に始動し、ホストが入力に対応できるようになると直ちにPCの応答時間を与え(in > end :)、pcの入力が完了すると直ちに、ホストの応答時間を与える(in < back :)。第3列は、コマンドが発せられてその応答があった日の時間を印刷する。

40 横の第1行は、コマンドを表示する。第2行は、PCまたはユーザがコマンドを発するのに要するタイマトビリオドを表示する。該時間は、ホスト・システムが入力に対応できるようになる時点から測定され、終了時間はユーザがエントリ・キーを押す時である。第3行は、システムが応答して付加的な入力に対する対応準備が整うまでに要するタイム・ビリオドを表示する。始動時間は、ユーザがエントリ・キーをタイプする時点からin > in定される。終了時間は、ホストが付加的入力を受けつけるようになった時である。

50 性能チャートの終わりでは、印刷された要約が、発せら

れたコマンドの数、平均() send : and (back :) 時間、最長(>5 and : and <back :) 時間、最短(>5 end : and <back :) 時間、および合計() send : and (back :) 時間を表示する。

1皿盆盈

システム上のユーザのターミナルにおいてシステム応答時間を判断することは、しばしば田辺な仕事である。通常は、試験を正しく行うために、ストップウォッチを買った人間がターミナル・オペレータをwt察するとともに、コマンドに関連する各システム応答の時間を計る。しかしながら、これでは時間がかかり、かつ不正確である。ユーザに関連する計時情報を収集することも困難である。

SESSIONを*P (性能チャート) オプションで用いることにより、タイマや特別のキーストロークでクロックを始動・停止する必要なしに、このような情報をを集め、かつセーブすることができる。性能オプションによれば、最初のシステム・アテンション・キー(ENTER, CLEAR1任意のPFまたはPA)から計時が開始される。

これらのキーの1つを押すとタイミングが取られるとともに、システムが、「使用可能」「入力は禁止されていない」というメッセージを戻すと、別のタイミングが取られる。この時点で、ユーザによるデータの入力が可能になる。真のエンド・ユーザ応答時間を決定することができる。ユーザがデータを入力する際、クロックは別のシステム・アテンション・キーが入力されるまで走り続ける。

別のシステム・アテンション・キーが入力されると、計時が再び始まる。そして、ユーザ間は、Pファイルにセーブされる。しかしながら、PCがコマンドをシステムに自動的に従わせること、およびユーザの介入なしでシステム応答時間の統計を集めることが好ましい。これを行う手段は、後で再試験に関連して記されている。

、Pファイルは、計時(性能)情報のすべてを保持している。該ファイルの終端における要約は、コマンドの数、最大のシステム・ラグ時間とユーザ・エンタリ時間、最小のシステム・ラグ時間とユーザ・エントリ時間、合計のシステム応答時間とユーザ入力時間、および平均のシステム応答時間とエンド・ユーザ入力時間を含む。個々のコマンドは行にリストされ、計時情報は列にリストされる。

システム性能テストが生成する性能チャートには、ヒューマン・ファクター・テストが生成するものと同じである。さらに、性能テストと性能チャートに現れ、計的のトリガとなる事象は、ヒューマン・ファクター試験に関連する前述の事象と同じである。

通1」1に五

再試験は、以前にテストした製品の試験を含む。

製品が一度試験されている場合は、該試験のある程度の記憶の存在が期待される。そのような記憶、は、PCの記憶であってよい。このように、PCは以前実行した試験を戻し、新しく得る結果を古い結果と比較することができる。また、PCを用いて、発見された問題をその発生場所も含めてドキュメントしたり、関連情報を繰り返し表示したりできる。PCを旧り、リースに対する新リリースのしきい試験モードで使うにせよ、性能しきいを越えた直後の製品のオンライン・テストで使うにせよ、または、リリースされてサポートが必要になった直後の製品のフィールド・エラーの通報に使うにせよ、本発明のシステムは、人間試験者よりもこれらの試験をより迅速にかつ能率よく指揮することができる。

しきい(回帰試験)

リリースを重ねる度に、ソフトウェア製品は変化し、かつ高級になる。しかしながら、このような高級化においても、オリジナルの機能を依然としてサポートするとともに、試験を施しておくべきである。これは、しばしば回帰試験と呼ばれるものであり、新しい機能が試験された後で生じる。

SESSIONを使うと1回帰試験は、製品が人間試験を受けるよりも十分前に存在しなければならない最小限の受容可能機能のしきい試験となり得る。機能の最小限セットは、初期のリリースにつき以前に実行した試験によって決定される。

このような初期試験の際にSESSIONプログラムを用いたならば、自動試験ケースが既にしきいテストへについて存在している。

SESSIONのプレイバック・ファイル・オプション30、およびIDバラメータは、再試験に関連している。プレイバック・オプションは、人間オペレータがPCキーボードについているかのように、SESSIONがホスト・システムを駆動することを可能にする。しかしながら、PCがシステムを駆動する場合、オペレータのミスや遅れは存在しない。PCは、ボス1～が許す限りいくらでも速く実行する。

ホストをプレイバックするのに用いるファイルは、以前のリリースの試験を以前に記録したものである。これが、テスト・ケースを意味する。

+kDバラメータ(差異ファイル)バラメータを用いることにより、SESSIONは、コマンドをプレイバックするだけでなく、ファイル上のスクリーンに関して以前実行したものと現在プレイバックされて表示中のものとの比較も行う。発見された差異はどれも差異ファイルにセーブされ、後で人間がレビューする。セーブされるのは差異だけである。スクリーンの数に差がないならば、SESSIONは、単にコマンドをホストに出し、かつ差異を識別し続けるだけである。

実行例を用いると、次のようにSESSIONを呼び出してしきい試験を実行することが可能である。

SESSION 151L*D PLAYBACK
SESSIONは、該例について以前に記録されたコマンドを自動的に検索し、どんな差異も検出する。比較のために使うスクリーン・ファイルは、第2a～2p図に示される編集前のシーケンス・イメージ・ファイルである。

第7図には、再試験に用いるシステム構成が示されている。しきい試験が完了すると、直ちにレビュー・プログラム50による差異ファイル26のスキャンが可能になる。イメージ・ファイル26とコマンド・ファイル28もレビュー・プログラム50へ入力される。このプログラム50は。

以前のランからオリジナルのスクリーンを示し。

続いてそれに現在のラインの際に見つかった差異を重ねる。このような差異は強調すると、単一の文字の場合であっても認識が容易になる。特別情報は、・レビュー・スクリーンの底へ出てくる。そのような情報は、該スクリーンに到達するのに用いたコマンド、オリジナル・スクリーンのファイルの中のスクリーン番号、および差異を有するスクリーンの合計数を含む。

レビュー・モード

レビュー・プログラムによると、ユーザが、5ESSIONにおいて*Dオプションを持つファイルをコンソールがプレイバックした後、コンソールが扱ったデータを見れるようになる。オリジナルのユーザ・セツションが表示され、かつPCが発見したセツションにおける差異がオリジナルに重ねられる。ユーザは、オリジナル・セツションとPCラン・セツションの間で切り替えて、2つのランにおいて異なるスクリーンだけを見ることができる。

レビューによると、ユーザが保護（非入力）フィールドと無保護（入力）フィールドの差異だけを見ることが可能になる。このようにして、異なるシステムのある特性をマスクし、あり得る差異を無視することができる。エディタによっても、オリジナルのユーザが捕獲スクリーン上のマスク文字の生成が可能になる。5ESSIONはこれらのマスク文字を無視するので、かかる物理的なスクリーン位置で生じるどんな差異も続くプレイバック時には無視される。

レビュー・プログラム50はターミナル34に常駐し、ターミナル34によって呼び出すことができる。すべて5ESSIONプログラム12（第1図）によって生成されるイメージ・ファイル26、コマンド・ファイル28、および差異ファイル22は、別々にレビュー・プログラム50に入力できるし、どのように組み合わせて処理してもよい。

レビューは、申り、オプションを使ってプレイバック・モードで実行されつつある5ESSIONに依存している。レビューにおけるオリジナル・スクリーンから異なるスクリーンへの切替は、ます比較的遅く生じる。こ

れが直接的な全後続時間である。この結果、大きな差異のあるスクリーン上で、差異が突然現れることはないが、スクリーン上への表示には色が使われる。

レビュー・コマンドは次のような構文を持つ。

REVIEW <ファイルネーム> [, レベル] [アスタリスク] [分離記号] レベルは、DOSにおけるファイル・エクステンションと同様のものである。これによれば、使用中のファイルのデフォルトへのレベルが指定されたレベルにセラードされる。使用されないときのデフォルト・レベルは、そのファイルのタイプ（、C1、S、および、D）についての存在レベルの最高値となる。指定レベルは1から99の間の数であってよい。アスタリスク・パラメータは1分離スペースなしで直に、ファイルネームに続かねばならない。分離パラメータはファイルネームから分離されなければならず、かつ互いに少なくとも1スペース煎れていなければならない。2スクリーンPCを使う場合、カラー・オンリーは、レビューをカラー・スクリーンへ強制的に向ける。

例： REVIEW testfile 中12スクリーン・システムが使用中ならば、システム・スクリーンをどちらかのディスプレイに配置することが時に有用である。-1のために、レビュー・プログラムはカラー・スクリーンだけしか使うことができない。モノクローム・ディスプレイは、自由な状態のままである。

2スクリーン・コンソール・システムでの場合に、モノクローム・オンリーは、レビューの出力をモノクローム・スクリーンへ強制的に向ける。

例： REVIEW testfile 串0

20 コマンド・ファイルを確立するときは、ユーザ指定コマンド・ファイル・エクステンション番号が用いられる。例： REVIEW testfile C5このパラメータはレベル・パラメータのバリエーションであり、コマンド・ファイルだけについてファイル・レベルの指定を許す。

ユーザ指定差異ファイル・エクステンション番号は、差異ファイルを生成するのに用いられる。

例： REVIEW testfile h5このパラメータも、レベル・パラメータのバリエー

40 ションであり、差異ファイルだけについてファイル・レベルの指定を許す。

保護オンリー機能は、文字Pによって呼び出される。レビューは、保護フィールドで見つかった差異だけを表示する。

例： REVIEW testfile p保護フィールドで発見されたどんな差異も表示される。他のすべて（無保護フィールド）は、差異として表示されない。差異（、D）ファイルを生成中の間、このオプションで5ESSIONを実行することも可能である。その場合

50 、差異ファイルは保護フィールドの差異だけを位える。

そうでない場合、5 E S S I O N はすべての差異をセーブする。そして、それらを一掃するには、レビューを呼び出さねばならない。

クワイエリー機能は、プログラムの事前、事後のロゴおよび音声のはほとんどをオフにする。

例： RE V I E V testfile q プログラム・スタートおよびエンドのどのロゴも、ロゴのタイポマッチ・サウンドとともに抑止される。g がプログラムのファイルネーム・メニューから指定されたならば、プログラム・エンディング・ロゴだけが抑止される。これに対して。

イニシャル・プログラム・ロゴは既に表示杭である。無保護オンリー機能は、文字 U によって呼び出される。レビューは、無保護フィールドで見つかった差異だけを表示する。

例： RE V I E W testfile p 無保護フィールドで見つかった差異はどれも表示される。他のすべて（保護フィールド）は差異として表示されない。

差異（、D）ファイルを生成中の間、このオプションで 5 E S S I O N を実行することも可能である。その場合、差異ファイルは無保護フィールドの差異だけを備える。そうでない場合、5 E S S I O N はすべての差異をセーブする。そして、それらを一掃するには、レビューを呼び出さねばならない。

ユーザ指定スクリーン・ファイル・エクステンション番号は、スクリーン・ファイルの選択に使われる。

例： RE V I E W testfile s 5 このパラメータも、レベル・パラメータのバリエーションであり、スクリーン・ファイルだけについてファイル・レベルの指定を許す。

実施例では、下記第 10 表のようなコントロール・キーの組合せが最も有用であることがわかっている。

第 10 表

の差異の始まりに移す。

CTRL-E エンド・レビュー F 3 レビュー・プロゲラ戻る。

CTRL-END エンド・オフ・ファイル ユーザをすべてのファイルの終端へ移す。

CTRL-PGDN 後続スクリーン ユーザを後続の差異で CTRL-F 8 イスプレイへ移す。

CTRL-PGUP 先行スクリーン ユーザを先行する差異で CTRL-F 7 ディスプレイへ移す。

試験が H レビューされると、人間試験者は見つかった差異はエラーであるかについての判断をなすことが可能になる。

豆龜 U

自動試験は、しきい試験が完了した後に行われる試験を扱う。開発されたコードが、新しい機能を実行する責任

を持つ人間試験者の手にある。人間試験者が 5 E S S I O N の下で操作しており、かつ少なくともコマンドをセーブしている限り、ジョブをより一層容易にすることができます。

5 E S S I O N) 二よればコマンドとスクリーンのセーブが可能になり、その結果、人間試験者が容易にステップ列全体をリプレイすることができる。試験者がエラーを発見し、該エラーが開発者によって訂正されると 5 直ちにコードを再試験して修理を保証しなければならない

10 。この再試験では 1 元のエラーの原因となったステップを正確に繰り返すべきである。このようなコマンド列はしばしば覚えにくく、繰り返すのは時間の浪費である。人間試験者が元のエラーへ至るバスで他の何らかの変化が生じたことを知ることも困難である。

5 E S S I O N は 1 元のエラーへ至るケース 1 ローク列を正確にリプレイするとともに、しきい試験の場合のように実行された場合は、試験において該時点までに発達した差異にも言及する。自動試験では、5 E S S I O N を用いて試験者の骨折り仕事を減らすことができる。

20 コンバート・モード

コンバートは、5 E S S I O N によって生成されたファイルを人間の読解可能な形式に変更する翻訳プログラムである。コンバートは、人間が読めるファイルを 5 E S S I O N が読める形式に変更することもできる。

第 8 図には、イメージ・ファイル 2 6 、コマンド・ファイル 2 8 、およびインストラクション・ファイル 3 0 が接続されたコンバート機能 4 0 が示されている。ファイル 2 6 、 2 8 、 3 0 はまとめて、セーブされたセツション・ファイルと呼ばれ、参照番号 4 2 が付されている。

30 3 個のテキスト・ファイル 4 6 とホスト・フォーマット・イメージ出力ファイル 4 8 は、一括して参照番号 4 4 を付された人間読解可能ファイルを形成する。

人間読解可能ファイル 4 6 は、PC エディタを使って編集することができる。コマンド言語を使うと、新しいコマンドを挿入したり、古いコマンドを削除したり、ファイルを拡大・縮小したり、ファイルを使って全く新しいファイルを生成したりすることができる。5 E S S I O N が該ファイルを使用できるようにするには、コンバート 4 0 をもう一度実行しなければならない。

40 コマンド・ファイルは、2 つの方法のどちらかで生成できる。第 1 の方法では、5 E S S I O N プログラムを使ってコマンド・ファイルを記録する。ユーザは、5 E S S I O N の中のタスク開始定義コマンド (C L R I - M) を出す。すると、対話的なタスクが普通に行われる。該タスクが完了すると、タスク終了定義コマンド (これも C L R I - M) が入力され、最初に指定されていたタスクがコマンド・ファイルとしてセーブされる。コマンドを生成する第 2 の方法は、単にエディタを使い、以前定義したようなコマンドのリストを生成することである。続いてこのリストはコンバート・プログラムによっ

て処理され、実行可能なコマンド・ファイルになる。
 5 E S S I O N フォーマツト～に変換し直されたファイル42は、構文とロジックがチェック済である。これは、ベーシックやバスカルのプログラムをコンパイルするのに似ている。ここでの文脈において、コンパイルによる類推は、5 E S S I O N 読解可能コマンドと人間読解可能コマンドの両方を備えるファイルに関係する。エラーにはフラグが立てられ、重大性に応じて、変換は続行または終了される。ファイルは5 E S S I O N フォーマツトに直され、存在エクステンション番号（つまり1. Cナンバー1. Sナンバー、または、Hナンバー）の最高値に1を加えた値が与えられる。したがって、ファイルに重ね書きが行われることはない。
 コマンド・ファイルに対してコンパートを使う利点の1つとして、コンパートのサポートする言語は、5 E S S I O N が捕獲できるエミュレータ・キーを1つ以上許容することが挙げられる。これによれば、コマンド・ファイルが元々そう作られているよりも一層一般化されたタスクを達成することが可能になる。コンパイルの1つのタイプがファイルについてなされるので、スクリーン探索、セーブ、および可変パラメータと同様に、ルーピング制御がサポートされる。これらの言語コマンドのあるものは5 E S S I O N に限られているキーをシミュレートする。その他の言語コマンドは該言語だけに限られている。5 E S S I O N がレコーディング中に特殊言語制御を捕えることはできないけれども、コマンド・ファイルの形になった制御を理解することはできる。
 人間読解可能ファイルを5 E S S I O N 読解可能にすることに実行するコンパイルは、補正（コレクティブ）コンパイルの1タイプである。問題が修正点を過ぎてコンパイルされるか、または問題があまり難しいときだけ、コンパートはコンパイルを中止する。しかし、エラーがコンパイルの際に訂正可能だとわかると、該エラーは訂正される。
 必要なコマンドが落ちた場合、可能ならば、コンパートはそのようなコマンドを挿入し、5 E S S I O N の要求を満たす。このような付加されたコマンドは、元々人間読解可能形式から変換された5 E S S I O N 読解可能ファイルを再び人間読解可能形式に変換し直すときに目立つ。このような操作は、コマンド言語ファイルをデバッグするのに有用である。なぜなら、コンバイラの変更するコマンドが明白になるからである。このような二度の変換を行ったファイルを注意して検査すること、および二度目の変化は行わないことが薦められる。
 操作時に、コンパート・プログラマ40は、コマンド・ファイル28、イメージ・ファイル26、インストラクション・ファイル30、またはこれらの任意の組合せをテキスト・ファイル46に変換できる。さらに、イメージ・ファイル26は、コンパート・プログラム40によってホスト・フォーマット・イメージ・ファイル

48に変換可能である。
 コンパート・プログラム40を呼び出すには”。
 次のような構文を使わなければならない。
 C O N V [E R T <ファイルネーム> [f r o m
 ; 仁 o c l C c l E C n o n u m]] ここで、ブレケット[]でオプショナルであることが示されている前述のパラメータは、下記第11表の機能を表わしている。
 第11表
 10 f r o m c j t o c 変換の向き
 c f l 変換するファイルのレベル n o n u m
 カーソル・ロケーションを使わないコンパート・用いて、コマンド・ファイルを生成したり、修正したり、一膜化したりできるし、あるいは反復する自分を備えたホスト・セツションにすることもできる。
 王 j 二皿報
 エラー通報は、可能ならばコード開発者が問題を考えたり除去したりできるように、試験者およびユーザが問題をドキュメントする処理である。
 20 従来の文書化方法は、しばしばシステムをまずく表現していた。この結果、訂正が発効するまでの長い遅延、何が問題であるかについての混乱、および問題を決定する試みにおいて文書化されたエラーを越えて特別に通信を行う必要が生じる。
 さらに、ユーザは、エラー発見の際に使うステップ列を正確に覚えなくてよいから、エラーはしばしば正確に文書化され、開発者には無効であると思われてしまう6 SESSIONを使うことにより、ターミナル・セツションについて正確なレコードが保持される。
 30 このレコードはコマンドを含むだけではない。ユーザが問題に至るアクションをさらに記述するために同時期に付加することのできるテキストのただ1行だけでなく、ユーザが見るスクリーン全体も含む。これは前記のノート・ツ・スクリーン（N O T E T o 5 C R E E N）によッテ達成される。エディタ・プログラムを使うセツションが終了した後、ユーザ指定テキストのスクリーンまたはページの全体を入力できる。このような文書化によって、開発者は、ユーザが入力したコマンドだけでなく、ユーザがそれらを入力した時間、そしてその時にユーザが何を見て何を考えていたかを見る。
 40 ノロ1. タスキング・システム
 エミュレータ・オーバーレーとしての5 E S S I O N には、普通のエミュレータだけでは得られない機能がある。マクロは、P C レベルでの、B A T ファイル、またはホスト・システムでのC L I S T S もしくはE X E C 5 に似ている。5 E S S I O N を使って単にコマンド列を記録するだけで、マクロを生成することができる。このようなマクロはいってもブレイ・バックでき、マクロ・ファイル・ネームに至る反復的で難しいキーストロークを減らす。

本実施例では、下記第12表のコマンドがキーボードによって捕えられ得るすべてのコマンドの他に、後記のようにマクロをもっと柔軟つまり再構成可能にすることができる付加的なコマンドも含むセットを形成する。

第12表

キーボード・ファンクション

N 5 O F I

[I N S O N]

[P R T S C]

[E N T E R]

[T R C]

[I E E O I]

[E E O F C]

[H O M E C]

[B C]

[T A B]

C B T A B]

[L 1]

[D C]

[C H 3]

[I ()]

[L]

[E N D]

[P F 1]

[P F 2]

[P F 3 C]

[I " F 4]

[r ' F 5]

[P F G]

[P F 7]

[P F ' 8]

[P F 9]

[P F 1 0]

[P F 1 1]

[P F 1 2]

[P A 1]

[P A 2]

[C L E A R]

[D E L]

[B E E P C]

種変ベニ人二色

[x O]

[% 1 C]

[% 2 C]

[. 3]

[z 4]

[z 5 3]

鷗 6 C]

[z 7]

[% 8]

[\$ 9 C]

ストリングの りを、V I I 全

[F + C]

[F - C]

凡 p J 寸 I C 引エシ奇

[+ F]

q 世 d 1 で 9 飢?

10 [W A I T C]

キーが されるまで一止する

[P A U S E]

阻綾々務エ

[+]

[- C]

[C -]

[5 T O P]

G O T O のためのラベル

[: 0 C]

20 [: 1]

[: 2]

[: 3 C]

[: 4 C]

[= 5]

[= 6]

[= 7]

[= 8]

[: 9]

G O T O コマンド

30 [G O]

[G 1 C]

[G 2]

[G 3 C]

[G 4]

[G 5]

[G 6 C]

[G 7 C]

c G 8]

[G 9 C]

40 [N O P]

遅延ビヒリクαビエトオ井

[D [E L A Y]]

j システムのリリリチェック

C V M E

[M V S C]

ベ各スニ五辺区勤U

[L + I]

[L - C]

[B L N K +]

50 [I 3 I N に - C]

特" カーソル・ロケーション

[GOXY]

[5NAP]

[0LDXYC]

[(*)]

[傘)]

[TYPE R C]

[TYPE L]

[TYPE L1]

[TYPED]

リセツ1-G甲■

[R0C]

[R1C]

[R2]

[R3]

[R4C]

[115C]

[R6]

[R7C]

[R8]

[R9]

[READC]

Dm筐致

[VAR0C]

[VAR1]

[VAR2]

[VAR3C]

[VAR4]

[VAR5]

[VAR[i]]

[VAR7]

[VAR8]

[VAR9]

大文字／小文字

[CAPONC]

[CAPOF]

値をハにり当てる

[S[ETO+]

[5ET1+C]

[5ETZ+C]

[5ET3+]

[5ET4+]

[5ETS+C]

[5ET6+]

[5ET7+]

[5ET8+]

[5ET9+]

[5ET-C]

恩所又致匁先頭と の！なくともーの4. - - [5TR
IP]

コンバートおよびコマンド言語を使って、記録する代りにマクロを書込んだり、記録後に変更したりすることができる。コマンド言語によれば、条件法、ループ、ディスクへの書込み、可変パラメータ、および単に記録するだけでは複写できない他の多くの機能が許される。したがって、コマンド言語によれば、無限に処理する能力のある強力なマクロを生成することができる。一旦生成または修正された後、続いて編集可能な形のマクロ（、CMDファイル）がコンバート・プログラムの下で実行される。このため、ファイルは5ESS工〇N読が可能形式（、C#ファイル）になる。該ファイルは、最も複雑なマクロさえも正しくなるまでコンバートの下で何度も実行することができる。

下記第13表のマクロ・ファイルは、付加的なコマンドが挿入された結果、ファイルが一連のホスト・ファイル全体を更新することのできる簡単なマクロの1例である。FIND、GOTOおよびルーピング・コマンドのために、システム状態の変化に適切に反応できるという点で、該マクロは通常のマクロよりも柔軟なものになって20いる。

このファイルは、一連のファイルを更新するために、EDITを（VM上で）使うように設計されている。

= ズ

Xψ 門〇x ローロー

0 z 門-c l、l-0+ CL+CX-0 ()
H:) (ブ ロー O Lり ロー⁺
++ (コL-I uL+I -----

賞

ニ

30 エミュレーション

コンバートを用いると、マクロ・ファイルの一般化、長時間実行タスクまたは反復タスクの生成、毎日の操作のffff略化、およびユーザがホスト・システムと会話をするやり方のカスタマイズが可能になる。データ・セット・ネームや、後で可変パラメータに変換されるような特定入力で、マクロ・ファイルを捕えることができる。長時間実行タスクおよび反復タスクは、数個のキーストロークを入力し、ルーピングまたはブランチング・マクロでもって実行することができる。最後に、コンバートを用いると、ホストを登場させてコマンドとともにパラメータを受諾したり、編集が完了するとデータ・セットを自動的にセーブすることが可能である。例えば、データ・セットを編集するとともに、R8Q集が完了するとそれをセーブするコマンド・ファイルは、下記第14表のように現れる。

第14表

[CLEAR]

EDIT [%01 CUF、NTERI [1dAIT
コ . . .

50 [110MIE] SAVE CENTER] [H

OME] QUIT [ENTER] NTER I エディタを使うためのキーストロークは次のようになる。

t r l - P

EDIT MY-DATASET

MY-DATASETは [0%コ位置を占めており、システムは待機状態のエディタになっている。 I g 集が完了すると、ユーザは CL r l - W を押して該ファイルをセーブし、エディタを出ることができる。

E 0 発明の効果

本発明のタスキング・システムによれば、変化するホスト・システム環境に応答して様々なタスクを自動的に反復して規定、実行することができる。

4、図面の T f f 車な説明

第1図は、本発明の E S S I O N プログラムによる要素、および該要素にさらに付随的な要素も含めたものの間のデータの流れを示すブロック図である。

第2a～2p図は、実行例においてコンソール上に表示されるイメージを表わす図である。

第3図は、付随的要素を備えたエディタの構成を示すブロック図である。

20

第4a～4p図は、それぞれ第2a～2p図に関連するインストラクション・テキストを表わす図である。

第5図は、教育ドライバおよびそれを用いて操作できるモードのブロック図である。

第6a～6d図は、教育ドライバに関するエラーを表わす図である。

第7図は、本発明による再試験のための構成を示すブロック図である。

第8図は、本発明によるコンパイル・機能を示すブロック図である。

30

1

第1図

c 3

m :

○第3図

区

エ

一

=コ

サ

寸

派

凍

⑩ 日本国特許庁 (JP) ⑪ 特許出願公開
 ⑫ 公開特許公報 (A) 昭62-214437

⑬ Int.Cl.
 G 06 F 9/06 9/46 識別記号 310 340 庁内整理番号 C-7361-5B B-8120-5B
 ⑭ 公開 昭和62年(1987)9月21日
 審査請求 有 発明の数 1 (全33頁)

⑮ 発明の名称 再構成可能な自動タスキング・システム
 ⑯ 特願 昭62-27460
 ⑰ 出願 昭62(1987)2月10日
 优先権主張 ⑯ 1986年3月10日 ⑮ 米国(US)⑯ 838062
 ⑯ 発明者 ランダル・ヒュー・カーラー アメリカ合衆国ニューヨーク州リツチフォード、アール・デイ1、ボックス211番地
 ⑯ 発明者 ロバート・マーシャル・メスナード アメリカ合衆国ニューヨーク州エンディコット、ウイルマ・ストリート620番地
 ⑯ 出願人 インターナショナル・ビジネス・マシーンズ・コーポレーション アメリカ合衆国10504、ニューヨーク州 アーモンク(番地なし)
 ⑯ 代理人 弁理士 頤宮 孝一 外1名

明細書

1. 発明の名称 再構成可能な自動タスキング・システム
 2. 特許請求の範囲
 (a) 対話的に始動された操作を実行することでのきるホスト・システムと。
 (b) タスクの規定および実行を反復して行えるよういに選択されたインターフェーシング・プログラムがロードされた、プログラムを実行するための手段と。
 (c) 前記ホスト・システムと前記プログラムを実行するための手段とに接続され、前記ホスト・システムに対してコマンドを発するとともに、前記ホスト・システムからデータを受け取るインターフェースと。
 (d) 前記インターフェーシング・プログラムによって規定されたタスクに接続され、前記タスクが規定された後に生じた前記ホスト・システムの環境変化に前記タスクおよび前記インターフ

エーシング・プログラムを順応させる独立して実行可能な手段と
 からなる再構成可能な自動タスキング・システム。

3. 発明の詳細な説明

A. 産業上の利用分野

本発明は、タスクを生成、実行するためのシステムに関し、さらに詳しくは、動的に変化するホスト環境において使用するタスクを一般化するためのシステムに関する。

B. 従来技術およびその問題点

人間資源およびコンピュータ・システムの分野では、毎日の作業に因襲した反復的な作業に含まれる単調さを減らすことが求められている。例えば財務、科学または人事データを含むコンピュータ・レコードまたはファイルの更新、品質保証試験、編集操作等のタスクにおいて、繰り返される活動の量は膨大なものになる。このような単調な活動を人間が行うと、作業能率の低下、疲労、バーンアウト、そしてしばしば受け入れ難いほどの

特開昭62-214437(2)

エラー率の増加を招く。

外部環境変化が予測されない、最も繰返しの多い編集および試験の操作の際には、固定されたタスクつまりプログラム化されたルーチンが複数のファイルを更新するのに適当である。しかしながら、予測しなかつた状況に応じて変化することを求める場合が、対話的な環境では数多くある。本発明が対象とするのは、後者のタイプの状況である。

ホスト・システム環境の動的変化には、次のようなものが含まれる。

- a. システム情報メッセージ (つまり、HSM、階層記憶管理プログラム)
- b. エラー状態および因縁したシステム応答 (つまり、プログラム異常終了)
- c. ささやかなものから破滅的なもの (つまり、システム・クラッシュ) までのシステム障害

従来、タスクは、変化しない環境状況、つまり環境の変化の仕方が予め予想されるいくつかのうちの1つに収まる状況に制限されていた。この結

果、変化する環境に応答して様々なタスクを単独で実行することができるロボットつまりプログラムの代りに、各々が单一のタスクだけを実行するロボットつまりプログラムが増えてきた。

変化を予測できない環境の性質ゆえに、そのような環境において適切に実行されるようにするにはいかにタスクを修正すべきかを事前に知ることができない。したがって、タスクを修正可能にするには、操作環境での予想されない状況を識別するとともに、あるモードで学習つまり獲得した経験を別のモードで実践する能力を備えなければならない。

本発明の目的は、人手を介さずに、反復してタスクを実行するシステムを提供することにある。

本発明の他の目的は、反復的なタスクを実行でき、かつ変化する環境条件に適合するシステムを提供することにある。

本発明のさらに他の目的は、対話を用いた環境における動的変化に順応するように、自動的に再構成することが可能なシステムを提供することにある。

る。

C. 問題点を解決するための手段

本発明によれば、再構成可能な自動タスキング・システムが提供される。ホスト・システムは、対話的に開始された操作、つまりタスクを実行することができる。インターフェーシング・プログラムのロードされた、プログラム実行用装置が提供される。該インターフェーシング・プログラムは、タスクの規定と実行を反復するように適合している。該インターフェーシング・プログラムは、タスクの規定と実行を反復するように適合している。該インターフェースが接続され、該ホストに対してコマンドを発するとともに、該ホストからデータを受け取る。該インターフェーシング・プログラムによつて規定されたタスクには、独立した タスク・モード が接続される。この独立したプログラムは、タスクが規定された後、該インターフェーシング・プログラムに一般性を加える。

D. 実施例

コンピュータ環境向けに意図された本システムは、次の3つの基本操作モードからなる。それは、教育 (エデュケーション)、試験 (テスティング) および再試験 (リテスト) である。

これらの基本操作モードのそれぞれは、以下で述べるようなサブ・モードからなる。システムの要は SESSION (セッション) プログラムである。これを用いることにより、セッションつまりタイム・ピリオドが前述の操作モードの1つに従つて指定されたタスクの実行に費される。SESSION プログラムとセッション (タイム・ピリオド) のこのような違いを、本明細書では前者を英語のままで使うことによつて表現している。

第1回には、本発明によるシステムの要構造と、要構造のデータの流れが示されている。

IBM (登録商標) 社製のデータ通信アダプタ・カード付モデル PC / XT のような、コンソール、すなわちいわゆるスマート・コンピュータ・ターミナル 10 が、コンピュータ・プログラムのロー

特開昭62-214437(3)

ドと実行に適用される。コンソール10は、英数字とグラフィックスを表示する、好ましくは解像度の高いディスプレイ、つまりカラーのブラウン管を含む。本実施例では2つのディスプレイ・スクリーンが用いられるが、これらは、一方はカラーで情報を表示でき、他方はモノクロームで情報を表示できるという事実によつて区別される。本明細書では、ディスプレイをしばしばモニタまたはスクリーンと呼び、表示される情報をスクリーン・イメージと呼ぶ。SESSIONと呼ばれるコンピュータ・プログラム12は、コンソール10に専用する。SESSIONプログラムの持つ機能は多く、そのすべては独立に用いることもできるし、他と関連させて使うこともできる。

コンソール10は、ホスト・コンピュータ14との通信、つまりホスト・コンピュータ14との間でデータのやりとりを行うことができる。ホスト14の例は、IBMモデル3081プロセッサのようなメイン・フレーム・コンピュータである。

ホスト14、コンソール10、および下記の様

なファイルの間の通信は、SESSIONソフトウェア12によつて行われる。ホスト・コンピュータ14とコンソール10の間のそのような通信を可能にする機構がそこにはある。そのような機構の1つにエミユレータと普通呼ばれるものがあるが、これはIBM社からPC/VMボンド・システムとして入手できる。

図面で一括して参照番号15で示されているのは、ユーザの選択可能な入力ファイルであり、SESSIONプログラム12に接続されている。例えば、イメージ・ファイル16は、システムで下記のように使われるスクリーン・イメージを備える。別のファイル18はコマンドを記憶し、さらに別のファイル20は計時または性能データを記憶する。これら3個の入力ファイル15は同時に使用できる。記述を簡単明瞭にすめために、第1図ではイメージ・ファイル16、コマンド・ファイル18、および性能ファイル20をそれぞれ单一のファイルとして示しているが、条件が許すなら本発明のシステムではそれぞれ複数個のフ

イルで構成してもよい。実施例の記載は本発明の範囲を限定するものではない。必要に応じてファイルをどのように組み合わせて構成してもよい。

出力ファイルと呼ばれ、図面では一括して参照番号21で示される第2のファイルのセットは、SESSIONプログラム12が生成したデータを受け取るためのものである。差異ファイル22は、以前に記憶されたデータを最近生成されたデータの会話セットと比較して得られる情報を記憶する。この情報は、オペレータまたはユーザが下記のように再試験モードで再生機能を実行するときに最も有用であるが、教育モードでスクリーンを更新するのにも使うことができる。

出力性能データ・ファイル24は、システムと人間であるユーザの両方の、性能つまり効率の記録を備える。性能データ・ファイル24は、後でアクセスし、かつ、それに統いて解析することの可能な計時情報を備える。

出力イメージ・ファイル26もSESSIONプログラム12によつてアクセス可能であり、コ

ンソール10に表示されるスクリーン・イメージを表わすデータを備える。

SESSIONプログラム12に接続された出力コマンド・ファイル28は、セッションの間にオペレータが出すコマンドを記憶する。

SESSIONプログラム

SESSIONプログラム12は、コマンドとスクリーン・イメージ全体の両方を捕えることができる。わかりやすくするために、本発明の主要な機能のいくつかをオーディオまたはビデオ・テープレコーダの特定の機能に例え、「レコード」「再生(プレイバック)」のような通常のレコーディング機能に関連した用語を使うことにする。キヤブチヤリング(捕獲)またはレコーディング(つまり、PCにおける大容量記憶データ)が起る時点から、SESSIONそれ自身を含むいくつかのプログラムの1つを走らせることができる。これらのプログラムは、エジュケータ、エディタ、コンバータ、およびレビューである。

第1図に示されていない人間のユーザは、コン

特開昭62-214437(4)

ソール10を通してホスト14に常駐するシステム・プログラムをログオンする。続いて、ユーザはトグリングによってコンソール10にアクセスし、コンソール10に前からロードされているSESSIONプログラム12の実行を開始する。SESSIONプログラム12は、ユーザをホスト14に再接続する。SESSIONプログラムには、通常のキーボード(図示せず)または他の適当な手段によってPCコンソール10から呼び出すことができる。

SESSIONという単語と適当なパラメータが入力される。コマンドSESSIONについての補文は、次の通りである。

SESSION <ファイル名> [s=p=d+n=a ve sd pl df cf
ステップ セイブオン/セイブオフ 0/1 再生 マスク=h]

ここで、括弧内[]でオプショナルであることが示されている前述のパラメータは、下記第1表の機能を表わしている。

第1表

使用するファイルのレベル。

システムに対するコマンドを生成するのを助けるのに最も有効なことは、特定コマンドを実行するため、特定のファンクション・キーをキーボード上に定義することである。SESSIONプログラム12においては、このようなファンクション・キーがデフォルトによって下記第2表のように定義されている。しかしながら、ユーザが通常のPCプロファイル・プラクティスに従つて、コントロール・キーをどのように定義することも可能であることを理解しなければならない。以下のファンクション・キーの定義はデフォルト・システム定義であり、一般的なアプリケーションにとって最も有用であることがわかっている。

第2表

キー	ファンクション	説明
CTRL-B	ブランク・スクリーン	別のCTRL-Bがスクリーン・イメージを回復するまで、スクリーン・イメージを消す。
CTRL-D	デイレクトリ	指定されたディレクトリ・マスク・ファイルをリストする。

* s スクリーン・イメージ・ファイルをセーブする。

* p 性能ファイルをセーブする。

* d 差異ファイルをセーブする。

* n 同じファイルに重ね書きする。

* a 既存のファイルに追加する。

VM VM(仮想記憶)システム・チエッキングを含む。

s # .s エクステンションを使う。

p # .p エクステンションを使う。

d # .d エクステンションを使う。

o # .o エクステンションを使う。

ステップ 各ユーザ・コマンドの再生と、それらの間でのボーズ。

セイブオン/セイブオフ 第1のスクリーンからレコード・モードを入力する(または入力しない)。

0/1 システム・スクリーンを、分離した白黒またはカラーのモニタに表示する。

再生 再生モードでSESSIONを入力する。

マスク=n ユーザ指定の取消し文字(n)。差異ファイルを作成するときに用いる。

CTRL-E エンド SESSION SESSIONプログラムを終了させ、DOSへ戻る。

CTRL-F1 ヘルプ SESSIONキーを定義するヘルプ・ウインドウを表示する。

CTRL-I 直接セーブ 現在のスクリーンをディスク上のスクリーン・ファイルへコピーする。

CTRL-L ロック・キーボード パス・ワードに基づくキーボード・入力を禁止する。

CTRL-N マクロ・ファイル マクロ(コマンド)ファイルの開始と終了を切り替える。

CTRL-N ノート・ツ・スクリーン 70文字のノートをスクリーンと共にセーブすることを可能にする。

CTRL-P マクロ・ファイル プレイバック コマンド・ファイルの実行を始める。

CTRL-Q クイット& プレイバック・ファイルの実行をやめて、キーボードをユーザに返す。

CTRL-R レコード・トグル セイビング・オンまたはオフを切り替える。

CTRL-S ステップ・モード 一度に1つのキーストロークでプレイバック・コマンドへ切り替える。

特開昭62-214437(5)

CTRL - V	VMトグル	VMシステムが使用中であることを明確にする。	に表示されるスクリーンを下記第3表に示す。なお、各キーに対応するファンクションが英語で示されているが、その訳および意味は上記第2表の通りである。
CTRL - V	ウェイト	一時的にプレイバックを中止(プレイバック)するように切り替え、ユーザ入力を許す。	
CTRL - X	イグジット	PCオペレーティング・システムへ出る。ユーザがEXITをタップすると、SESSIONへ戻る。	
ALT - R	タイプ・ライト	普通のタイピングを指定する。	
ALT - L	タイプ・レフト	キーがタイプされた後でカーソルが左へ動くことを指定する。	
ALT - D	タイプ・ダウン	カーソルが下がる。	
ALT - U	タイプ・アップ	カーソルが上がる。	
ALT - O	モノ・オンリー	モノクローム・スクリーンだけを使う。	
ALT - I	カラー・オンリー	カラー・スクリーンだけを使う。	
ALT - 2	モノ・アンド・カラー	モノクロームとカラーの両方のスクリーンを使う。	

SESSIONプログラムの中からキー Ctrl - F1 を押してヘルプ・メニューを呼び出したとき

第3表

EMULATOR FUNCTIONS

Enter Caps-Lock Clear	:	Scroll-Lock	:	Ctrl-Break				
e78	MVS	SESSION FUNCTIONS						
Blank Screen	:	Ctrl-B	Type Right	:	Alt-R	Playback	:	Ctrl-P
Directory	:	Ctrl-D	Type Left	:	Alt-L	Quit Play	:	Ctrl-Q
End Session	:	Ctrl-E	Type Up	:	Alt-U	Record	:	Ctrl-R
Instant Save	:	Ctrl-I	Type Down	:	Alt-D	Stop Mode	:	Ctrl-S
Lock Keys	:	Ctrl-L	Mono Only	:	Alt-O	VM Checks	:	Ctrl-V
Make File	:	Ctrl-M	Color Only	:	Alt-I	Wait	:	Ctrl-W
Note	:	Ctrl-N	Mono & Color	:	Alt-Z	Exit DOS	:	Ctrl-X

SESSIONプログラム 1.2 は、プログラム呼出し時に総括的な(ゼネラル)ファイルネームを指定することにより、周期的に(つまり毎日)走らせることができる。このようなファイルネームは、セッションに関する重要な情報をセーブする必要ができたときにセーブ・スクリーンズ・オプションを指定した現在の日付であつてよい。特定情報をセーブするために、Ctrl - M(メイク・

マクロ・ファイル)オプションが用いられ、該情報に関連してあるファイルネームが指定される。この結果、ユーザがマクロ・ファイルをそのまま直後にプレイバックできるようになる。

セッションのレコーディングがその名の下で行われるファイルネームが、呼出し時に指定される総括的な SESSION ファイルネームであるならば、該 SESSION ファイルネームが変更されるまで、ファイルのプレイバックは不可能である。ファイルネームを変更するには、Ctrl - E で SESSION を終わらせ、かつ違うファイルネームで SESSION を再始動する必要がある。なぜなら、呼出し時に生成される SESSION ファイルは、出力用に開かられるファイルであり、入力と出力の両方として使うことはできないからである。したがつて、その日その日のルーティン・オペレーションについては、現在の日付(例えば、AUG 27)のような属性的な(ジエネリック) SESSION ファイルネームを用いることが望ましい。

特開昭62-214437(6)

本発明の内容を理解するために、続いて実行例を説明する。以下の例は説明の目的だけでとりあげものであり、これによって本発明の範囲が制限されるわけではない。以下の例は、学生に I S I L 言語のコマンドの使い方を教えるためのものである。この例では、"IBM Virtual Machine/ System Product: System Product Eclitor Command and Macro Reference" (Endicott, NY, Second Edition 1982) に記述されているよう、システム・エディティング・プログラム (例えば X E D I T) が必要とされる。

まず、ユーザは、次の PC コマンドを入力する。
S E S S I O N I S I L S

ユーザは、一旦レコーディングを開始したいポイントを決めると、コンソール 10 から C T R L - R を入力する。すると、システムはレコーディング・モードとする。* S が元の S E S S I O N コマンドで指定されたので、システムはキーストロークと、これから入力されるスクリーン・イメージを捕える。

～第 2 p 図に示されるように、与えられたスクリーンを凍結して獲得するために、著者がコンソールのアテンション・キーを押す。その後セッションを続けることができる。このように繰り返しアテンション・キーを使うことは、S E S S I O N を使う著者にとって余分な仕事ではない。なぜなら、ホスト・システムでの作業の際に、アテンション・キーを使ってシステム・レスポンスを得ることは普通に行われている。

E N T E R (入力) キーはアテンション・キーの一例である。普通、アテンション・キーは、スクリーンの更新またはリフレッシュに使われる。アテンション・キーの他の例は、C L E A R 、 P A 1 、 P A 2 、およびファンクション・キー F 1 ～ F 1 2 である。

次に、著者はコマンド I N P U T を入力して、X E D I T 入力モードを呼び出す。結果として生じて記録されるスクリーンが第 2 c 図に示されている。著作が次に入力するステートメントは、

: p. Paragraph [CR]

C T R L - R が入力された後、第 2 a 図に示される最初のスクリーンは、新しく名付けられたファイル I S I L に記録される。I S I L D E M O S C R I P T というファイルネームを持つ V M データ・セットを生成するために、ユーザは次のコマンドを入力する。

X E D I T I S I L D E M O S C R I P T A I .

このステートメントにより、生じつつあるセッション会話を忘れて、ユーザが X E D I T システム・エディティング・プログラムを実行できるようになる。したがって、ユーザが單に X E D I T ルーティンを実行し、X E D I T 環境で操作するだけで、レコーディング・セッションがユーザにとって透過的に実行される。結果として生じて記録されるスクリーンが第 2 b 図に示される。

この実施例では、著者が、学生に文字 h 1 ～ h 6 によって識別されるサブ・ヘッディングまたはヘッダの使い方を教える教育パッケージを生成する。これらの文字および同様の文字は、タグとして知られている。時々このセッションでは、第 2

used like :p.This

である。ここで、[C R] はキャリッジ・リターン・キーを表わしている。文字 : p . は、新しいパラグラフの始まりを示す。これらの文字に続くすべてのテキストは、パラグラフの一部として受け入れられる。結果として生じて記録されるスクリーンが、第 2 d 図に示されている。

著者が次に押すキーは、単に E N T E R だけである。この結果、著者が後の修正のために前のスクリーン・イメージをセーブできるようになる。したがって、今は、第 2 d 図に示されるスクリーン・イメージのコピーが 2 つ存在する。第 2 e 図に示されるスクリーンを参照することにより、学生はテキスト上の I S I L タグの効果を直接見ることができる。今度現われるスクリーンは、第 2 f 図に示される通りである。

続いて、著者は、

: h1. Head level one [CR]

: h2. Head level two

をタイプして入力する。

特開昭62-214437 (7)

文字 : h 1. および : h 2. は、テキストの中のヘッディングとサブヘッディングを示す。ヘッダ・レベルを示す数字は、1 から 6 までである。結果として生じて記録されるスクリーンが第 2 g 図に示されている。

続いて、著者は、

```
: h3. Head level three [CR]
: h4. Head level four [CR]
```

をタイプして入力する。結果として生じて記録されるスクリーンが第 2 h 図に示されている。

続いて、著者は、

```
: h5. Head level five [CR]
: h6. Head level six
```

をタイプして入力する。結果として生じて記録されるスクリーンが第 2 i 図に示されている。

著者が ENTER キーを再び押して第 2 j 図の複数スクリーン・イメージをセーブすると、第 2 j 図に示されるように、後で該イメージを変えてテキスト上の ISL タグの効果を示すことが可能である。

からである。今度現れるシステム・スクリーンは、第 2 n 図に示す通りである。

続いて、著者は、

```
: ul. [CR]
: li. item1 [CR]
: p. stuff for 1 [CR]
: li. item2 [CR]
: eul.
```

をタイプして入力する。

文字 : ul. は、無秩序リストの始まりを示し、文字 : eul. は同リストの終りを示す。結果として生じて記録されるスクリーンは、第 2 o 図に示される通りである。

続いて著者が ENTER を押すと、第 2 p 図に示されるように XEDIT INPUT モードが終了する。

ユーザは、レコーディングを終えると、CTRL - R を入力してレコーディング・セッションを終了する。しかしながら、XEDIT システム・プログラムの実行は依然続いている、翌むならユ

ニ度のシステム・スクリーンは、第 2 r 図に示す通りになる。

リストを生成するために、著者は次のテキストを入力する。

```
: sl. [CR]
: li. item1 [CR]
: li. item2 [CR]
: esl.
```

文字 : sl. は、単純（シンプル）リストの始まりを示す。文字 : li. は、リストの項目つまりメンバを識別する。文字 : esl. は、単純リストの終りを示す。結果として生じて記録されるスクリーンは、第 2 l 図に示されている。

テキストの ISL タグの効果を見るために、著者が再び ENTER キーを押すと、複数スクリーン・イメージが生じるが、これは後で変更されて第 2 m 図に示されるスクリーンを生じさせる。リスト（項目 1 と項目 2）がレベル 6 のサブヘッディングの下に現れる。なぜなら、該レベルは、リストが開始する前に書かれた最後のレベルだ

ユーザは普通はそこから出ることができる。また、レコーディング・モードが実現されていない間でも、ユーザは依然として SESSION プログラムを実行している。SESSION プログラムを終えるために、ユーザは CTRL - E をタイプする。

SESSION が実行中である間に、2 つのファイルが作成された。一方はスクリーン・イメージ 26 (第 1 図) についてであり、他方はコマンド 28 についてである。

セイブ・スクリーン・イメージのエディティング (編集)

ここで、著者は以前に記録したあるスクリーン・イメージを修正するとともに、自動的に生成される基礎インストラクション・テキストにヘルプ・テキストを加え、ヘルプ・ファイルを生成したいと思うかもしれない。このようなことを行うために、次の構文でエディタが呼び出される。

EDITOR <ファイルネーム> [s# h# c# 0/1]

上記簡略化したパラメータは、SESSION

特開昭62-214437 (8)

コマンドに関して記述したもの（前記第1表）と同じ意味を持っている。

さて、著者は、次のコマンドを入力して実行例を進める。

EDITOR ISIL

エディタ機能では、ユーザの定義可能な特定のキーが、デフォルトによって下記第4表のように割り当てられている。しかしながら、ユーザは、通常のパーソナル・コンピュータ・プラクティスに従つてどのようにキーやキー・コンビネーションを定義できることを理解すべきである。

以下で述べるのは、エディタ・プログラムで使われるキーストロークのリストである。これらのキーは、すべて Ctrl(コントロール) キーまたは Alt(代替) キーと一緒に使われる。すべての定義の形式は、

Ctrl- <key> または Alt- <key>
であり、Ctrlと他のキー、または Altと他のキーを同時に押す必要があることを示している。これらのキーは、エディタのヘルプ・パネルの中で

も定義される (Ctrl-F1参照)。

第4表

キー	ファンクション	説明
CTRL-A	属性セレクト	タイプされた文字のフィールドの型。
CTRL-B	バックグラウンド・カラー・セレクト	フィールドのバックグラウンド・カラー。
CTRL-C	コマンド入力	コマンド・ファイルの修正/生成を許す。
ALT-C	コピー・マーク	マークされた領域をカーソル位置へコピーする。
CTRL-BACKSPACE	ラインの削除	ディスプレイから現在のラインを削除する。
CTRL-D	デイレクトリ・リスト	指定されたデイレクトリ・ファイルを表示する。
CTRL-E F-3	エンド・エディタ	エディタ・プログラムを終了させ、DOSへ戻る。
CTRL-X	イグジット・エディタ	PCオペレーティング・システムへ出る。ユーザが EXITをタイプすると、エディタへ戻る。

CTRL-END	スクリーンのトップ	カーソルをスクリーンのトップへ動かす。	ESC	セイブ・スクリーンズ	ヘルプおよびシステム・スクリーンをセーブし、次のスクリーンを表示する。
ALT-F4	ラインの終りまでの消去	カーソルからラインの終りまで文字を消す。	CTRL-CRLF	・スプリット・ライン	カーソルの所で現在のラインを分ける。
CTRL-G	ゴージ・モード	エディタをゴージ(GOTO)モードにする。	F9	スワップ・スクリーンズ	活動スクリーンをエディットに変える。
F1	ヘルプ	エディタ・キーを定義するヘルプ・ウインドウを表示する。	ALT-U	アンマーク	コピーまたはムーブのためにタグのついた領域のタグを取る。
ALT-J	ラインの接合	近くライン上のテキストを現在のラインの終りにつなぐ。	ALT-R	タイプ・ライト	普通のタイピングを指定する。
ALT-L	マーク・ライン	コピーまたはムーブ(移動)用のラインを付ける。	ALT-L	タイプ・レフト	キーがタイプされた後でカーソルが左へ動くことを指定する。
ALT-M	ムーブ・マーク	マークされた領域を現在のカーソル位置へシフトする。	ALT-D	タイプ・ダウン	カーソルが下がる。
PGDN F8	後続スクリーン	ユーザを次のシステム/ヘルプ・スクリーン・ディスプレイへ移す。	ALT-U	タイプ・アップ	カーソルが上がる。
PCUP F7	先行スクリーン	ユーザを先行するシステム/ヘルプ・スクリーン・ディスプレイへ移す。	ALT-O	モノ・オンリー	モノクローム・スクリーンだけを使う。
			ALT-I	カラー・オンリー	カラー・スクリーンだけを使う。
			ALT-Z	モノ・アンド・カラー	モノクロームとカラーの両方のスクリーンを使う。

特開昭62-214437(9)

CTRL-HOME スクリーンの底 カーソルをスクリーンの底へ移す。

F1 を入力して呼び出されるエディタ・ヘルプ。スクリーンは、第5表の通りである。なお、各キーに対応するファンクションが英語で表示されているが、その訳および意味は上記第4表の通りである。

第5表

Editor Help

-LINE CONTROLS-		-MARKING-	
Alt-F4	: Erase EOL	Alt-C	: Copy Mark
Alt-J	: Join	Alt-L	: Line Mark
Ctrl-CR/LF	: Split	Alt-M	: Move Mark
Ctrl-Backspace	: Delete Line	Alt-U	: Unmark
-SCREEN CONTROLS-		-SPECIAL COMMANDS-	
F1	: Help	Ctrl-B	: Background Color Select
F3	: Exit Without Saving	Ctrl-C	: Command Entry
F7/PgUp	: Show Previous Screen	Ctrl-D	: Directory Listing
F8/PgDn	: Show Next Screen	Ctrl-E	: End Editor
F9	: Swap Screens	Ctrl-G	: Goto a Screen
ESC	: Save Screen and Continue	Ctrl-N	: Note Display on Screen

は、個々のヘルプまたはシステム・スクリーンの中だけでなく、両者の間でも実行することができる。

また、エディタは、カラーおよび属性の情報の定義を可能にする。背景と前景の両方の色を変更できるし、入力領域も同様に定義できる。

第3図には、ターミナル34に接続されたエディタ38が示されている。エディタ38に入力できるファイルは、以前に生成されたイメージ・ファイル26、コマンド・ファイル28、およびインストラクションつまりヘルプ・ファイル30である。

エディタ38は、新しいイメージ・ファイル26a、コマンド・ファイル28a、およびインストラクション・ファイル30aを生成することができる。エディタ38は、そのようなファイル26a、28a、または30aの何れをも、それぞれ既存ファイル26、28、および30からか、またはそれらに関係なしに、生成することができる。

Press any key to continue

エディタ機能

エディタは、SESSIONで生成されたファイルと用いられるスクリーン指向の編集プログラムである。エディタを、スクリーン・ファイル(.S#)、ヘルプ・ファイル(.H#)、そしてコマンド・ファイル(.C#)にさえも用い、新しい情報の修正、生成の少なくとも一方を行うことができる。

エディタは、システム・スクリーンをカラー・モニタに配置するとともに、ヘルプ・スクリーンをモノクローム・モニタに配置する。デフォルトの場合、編集可能な領域はヘルプ・スクリーンに向かう。單一スクリーン・システムではヘルプ・スクリーンが表示され、トグリングによってシステム・スクリーンがアクセスされる。

エディタは通常の編集機能を多く持つている。領域をマークして移動、コピーしたり、ラインを挿入、削除することが可能である。これらの機能

エディタは、スクリーンおよびコマンド・ファイルの存否に関係なく使用できる。SESSION捕獲ファイルと組びつけて使うのが最も有効であるが、エディタはSESSIONから独立して使うことができる。エディタによって生成または修正されたファイルを、再びSESSIONに入力したり、またはエンジュケータに入力したりできる。

ヒューマン・ファクターのテストにおいて、エディタはプロトタイピングのツールである。システム全体を、普通のテキスト・エディタのスピードで模倣することができる。これらのスクリーンは、視覚的プロトタイプとして使えるだけでなく、エンジュケータ・プログラムがアクセスするときの基礎的なプロトタイプとしても使える。

エディタは、エラーをドキュメント(文書化)するのに使える。挿えた問題が大きすぎてどのスクリーンにもノートの形で記述できない場合、エディタを用いて適用可能なシステム・スクリーンについて全部のスクリーン・テキストを生成して

特開昭62-214437 (10)

もよい。

エディタを用いて、テスト・ケース（コマンド・ファイル）の修正もできる。もつとも、この目的のためには、コンバート（CONVERT）が下記のように設計されている。

どのエディタの中でもスクリーンとヘルプを交換して使えるけれども、エディタ・プログラムを用いると、これらのスクリーンを生成するとともに、それらが提示されるやり方でそれらをながめることができます。エディタは、コマンド・ファイルとともに、ヘルプおよびスクリーン・ファイルの両方を修正することができる。これがSESSION捕獲教育パッケージにヘルプを付加する最も簡単な方法である。

エディタに特有のことではないが、エミュレーション・シールはその用い方を通して説明することができます。マクロ・ファイルを使ってスクリーン・ファイルを生成することにより、操作方法の全体の文書化、および教育の基礎の形成が可能になる。マクロにおいて教育の目的で急速に発生し

た情報テキストを表わす。第4a、4p図は、エディタおよびエミュレータ・プログラムによって自動的に生成される基礎インストラクション・テキストを表わす。

ここで、第4a～第4p図のスクリーンに記された英文の大意を記す。

まず、各スクリーンの先頭に共通して表示されているKeystrokes entered to get here: の次の行には、各スクリーンの1つ前の段階のスクリーンで入力されたキーストロークが示されている。また、各スクリーンの末尾において、Keystrokes needed for next screen: の次の行、またはtype、press の指示の直後には、各スクリーンの次の段階のスクリーンへ進むために入力すべきキーストロークが示されている。したがつて、例えば第4a図から第4b図に移るためには、xedit isildemo script 31とENTERを入力しなければならない。

第4a図は、「A」ディスクにISILDERSCRIPTというデータ・セットを生成す

すぎる複雑なタスクをSESSIONの中で挿入することができる（プレイバック*S）とともに、記述テキストを付加することができる（エディタ）、ユーザ・ベースでマクロを実行することができる（エミュレータにおけるユーザ・モード）。

さて、著者はCTRL-Gを押してGOTOモードを呼び出す。実行例で既に記憶済のものの中から最初に修正されるスクリーン・イメージは、イメージ・ファイルのページ5にある（第2c図）。著者は、CTRL-G5をタイプすることにより、第2d図の複製イメージをアクセスする。次いで、普通の編集機能を用いて該スクリーンを修正し、第2e図に示されるイメージに到達することができる。ここでユーザは、ESC（エスケープ）キーを押すことにより、第2d図のこの修正されたバージョンをセーブする。

第4a～第4p図には、第2a～第p図に示されるイメージにそれぞれ関連・対応するテキストを含むヘルプ・スクリーンが示されている。第4a～第4n図は、編集セッションの際に入力され

るために必要なコマンドの入力を求めている。第4b図は、標準的なXEDITスクリーン（第2b図）が表示されている状態で、入力モードを入力することを求めている。第4c図は、:p. コマンドが新しいパラグラフの始まりを示すことを述べるとともに、:p. コマンドを実際にタイプすることを求めている。第4d図は、ISILがこのタグをどう扱うかを見てみようと述べている。第4e図は、XEDITで使うテキスト・ファイルのフォーマットが書かれたISILファイルのフォーマットとほとんど関係がないこと、:p. タグによってテキスト内での分離とライン・スキップが生じることを述べ、最後にはエディタの再入力を求めている。第4f図～第4h図は、:h1. Head level one(ヘッド・レベル1)～:h6. Head level six(ヘッド・レベル6)の入力を求めている。第4i図は書かれたISILがどのように見えるか見ようと述べている。第4j図は、オンラインではわかりにくいけれども、レベル1のヘッドによって新しいページが該ヘッダをトリ

特開昭62-214437 (11)

ブにして印刷されること、レベル2のヘッドはボールド体のダブルストライクで印刷されること、レベル3のヘッドはイタリック体のダブルストライクで印刷されること、レベル4のヘッドはボールド体で印刷されること、レベル5はイタリック体で、かつテキストとインラインで印刷されること、レベル6のヘッドはイタリック体、そしてテキスト・サイズで、かつテキストとインラインで印刷されることを述べている。第4k図は、ISIのリストには単純、無秩序、秩序の3種類があることを述べ、単純リストを見てみよう呼びかけている。第4l図は単純リストの説明（第2l図に関して既に述べた）である。第4m図の説明も第2m図に関して既に述べたものと同じである。第4n図は、違うタイプのリストを見てみようという呼びかけである。

スクリーン・イメージの修正バージョンをセーブした後、システムは次のスクリーン・イメージに進む。ここで著者は、第2i図で示されるように元々セーブされていたイメージの修正を望むか

もしれない。これを行うには、CTRL-G (GOTO) 10を押す。すると、第2i図のスクリーンを適当に修正することが可能になる。ESCキーを押すことにより、修正されたイメージは第2j図に示されるようにセーブされる。

同様に、ページ13に現れる第2l図の複数を修正して、第2m図に示されるようなスクリーン・イメージを作ることができる。

これで教育パッケージは完成した。該パッケージはフロッピー・ディスクにロードされたり、またはネットワーク能力を使って教育がなされる任意の場所に送られたりする。一旦受け取られた教育パッケージを、場所毎のバリエーションをもたらすエディタ・プログラムを使って修正することができる。

教育モード

教育モードにおいて、システムは、自動著作ツールである。教育の著者は、SESSIONプログラム実行時にホストを使っているオペレータである。記録されたスクリーンおよびコマンドのど

れもが教育文庫で有用となるように記憶される。SESSIONプログラムが実行を終える否や教育モードを入力することにより、抽象ホスト・セッションがインストラクション個人教授形式においてPCレベルでシミュレートされる。

第5図には、教育ドライバ・システムが示されている。イメージ・ファイル26とコマンド・ファイル28は、第1図に前述して既述した。ISIコマンドを教えるのに用いたセッションの実行例において、スクリーンつまりイメージ・ファイル26は、レコーディング・モード時のホスト・セッションの間に生成されたスクリーン・イメージのすべてのコピーを結局備える。これらのスクリーン・イメージは、單一のイメージ・ファイル26に引き続いで記憶される。同様に、コマンド・ファイル28も実行例のセッションの間に生成される。インストラクションつまりヘルプ・ファイル30も、教育ドライバ・システムで付随的に備えられている。

教育ドライバ・システムの要は、通常の多位置

モード・スイッチ33が常駐する教育ドライバ32である。ドライバ32がその一部をなすコンピュータ・ターミナル34は図示しないが標準的なコンソールであり、ユーザがプログラム・ドライバ32と通信するためのキーボード等の入力手段を含む。図面で一括して参照番号36で示されているのはモードであり、モード・スイッチ33によつてドライバ32に接続されている。モード・スイッチ33を適当にセットすることにより、各モードを一意的にアクセスすることができる。本実施例では、8個のモードを使うことができる。モード36a～36cは個人教授アプリケーションで、モード36d～36fは参照アプリケーションでそれぞれ用いられる。モード36g、36hはユーテリティノードキュメンテーション・モードである。

SESSIONプログラム12（第1図）により以前に生成されたものと同様の性能データ・ファイル24もドライバ32に接続されて、必要に応じ計時データを受け取る。計時データは、ある

特開昭 62-214437 (12)

事象とそれに対するキーボードを通じての人間の反応との間の時間間隔を表わす。この時間間隔が大きくなるほど、もちろん人間オペレータは無能である。

教育ドライバ32は、コマンド・ファイル28、イメージ・ファイル26、および存在すればヘルプ・ファイル30を追加し、これらのファイルを個人教授風に提示するエジュケータ・プログラムを含む。

エジュケータ・プログラムは、個人教授と訓練、オンラインイン・ヘルプ、提示および表示とエラー文変化をサポートする操作モードを持つ。

エジュケータ・プログラムは、著者すなわち教師のために教育パッケージを著作する会話形教育ツールである。SESSIONプログラム12

(第1図)を使ってホスト・システム14を挿入することにより、エジュケータ・プログラムは、該ホスト・システム14をシミュレートでき、さらに著作する必要なしで基礎インストラクション・テキストの提供さえも行うことができる。完全な

モード 開始モード。ユーザ、ステップ、オート、コマンド、リスト、ゴーツの1つ。

0/1 出力をモノ/カラー・ディスプレイへ強制的に向ける。

エジュケータ・コマンドで、. # が指定されていなければ、システムは未指定ファイルのそれれについて最も高いレベルを設定する。パラメータ・レベル (例えば4) は、使用中のファイルのデフォルト・レベルを指定されたレベル (例えば、第4レベル, s4, o4, h4) に設定する。使用されないときのデフォルト・レベルは、そのファイルのタイプ・コマンド (.C)、スクリーン (.S)、またはヘルプ (.H) の存在レベルの最高値となる。指定レベルは、1から99の間の数であつてよい。

個々のパラメータはファイルネームから分離されていなければならず、かつ、互いに少なくとも1スペース離れていなければならない。例を次にあげる。

EDUCATOR ISIL STEP 1

教育パッケージを生成するために、エディタ・プログラムは、上述のようにスクリーン1つずつのヘルプ・テキストを生成しなければならない。前述の実行例において、そのようなスクリーン1つずつのヘルプ・テキストは、エディタ・セツションの際にシステム・スクリーンの一部のために生成される。残りのシステム・スクリーンは、教育ドライバ32が生成する基礎インストラクション・テキストを持つ。

エジュケータ・プログラムを呼び出す構文を次に示す。

EDUCATOR <ファイルネーム> [.s] [c] [h] モード 0/1]

ここで、括弧内 [] でオプショナルであることが示されている前述のパラメータは、下記第6表の機能で表わしている。

第6表

- .# 使用するファイルのレベル。
- s# スクリーン・レベル・ファイルを使う。
- c# コマンド・レベル・ファイルを使う。
- h# ヘルプ・レベル・ファイルを使う。

上記エジュケータ・コマンドの例によれば、ユーザが実行例に関連してエジュケータ・プログラムを呼び出すことが可能になる。

カラー・オンリー (つまり、出力デバイス1) は、(例えば、2スクリーンPCで) カラー・スクリーンが使用可能ならば、エジュケータ・プログラムの出力をカラー・スクリーンへ向ける。2スクリーン・システムが使用されているならば、システム・スクリーンをディスプレイの1つに配置することが時には有用である。

モノクローム・オンリー (つまり、出力デバイス0) は、エンジュケータ・プログラムの出力をモノクローム・スクリーンへ向ける。

コマンド・ファイル (つまり、C #) は、コマンド・ファイルだけについてファイル・レベルの指定を許すレベル・パラメータのバリエーションである。最近のバージョンのコマンド・ファイル (バリュー・エクステンションが最高のファイル) を使う代りに、指定値を持つファイルが使われる。指定ファイルが存在しないならば、存在レベルが

特開昭62-214437 (13)

最高のファイルが使われる。

ヘルプ・ファイル（つまり、H#）も、レベル・パラメータのバリエーションであり、ヘルプ・ファイルだけについてファイル・レベルの指定を許す。

スクリーン・ファイル（つまり、S#）もまた、レベル・パラメータのバリエーションであり、スクリーン・ファイルだけについてのファイル・レベルの指定を許す。

オート（AUTO）モードが2スクリーン・システムで指定されると、システムは7秒間休止し、コマンドをタイプし、3秒間休止し、このような循環をしながら次のスクリーンへ至る。単一スクリーンでは、オート・モードは2秒間休止し、コマンドがタイプされた状態でシステム・スクリーンを3秒間表示し、このような循環をしながら次のスクリーンへ至る。ファイルの終りに達すると、該サイクルが再び第1のスクリーンで始まる。

コマンド（COMMAND）モードが指定されると、システムはユーザがコマンドをタイプするのを待

ち、統いて正確にマッチするものを求めてリスト・モードとともに使用可能なコマンドのリストを探索する。マッチするものが見つからなければ、入力されたストリングを備えるすべてのコマンドがマークされる。マークされたコマンドが1つだけならば、そのスクリーンがアクセスされる。1以上のコマンドがマークされたなら、マークされたコマンドのウインドウがユーザ・オプションとして表示される。マークされたコマンドがなければ、COMMAND NOT FOUND（コマンドが見つからない）メッセージが表示される。

ゴーク（GOTO）モードが指定されたなら、システムはユーザがスクリーン番号を入力できるウインドウを表示する。エミュケータ・プログラムは、指定されたスクリーン番号に分岐する。番号が教育パッケージのサイズよりも大きいならば、最後のスクリーンがアクセスされる。対応するヘルプ・スクリーンは、適当なゴーク指定スクリーンに連係される。

リスト（LIST）モードが指定されると、シ

ステムは教育パッケージ内のすべての使用可能なコマンドのウインドウを表示する。エントリ・タイプのキーは、コマンドとして表示されない。コマンドは、関連するスクリーン番号とともに表示される。ウインドウの中でリストをスクロールしてコマンドを選択することが可能である。選択されたコマンドは、教育パッケージの中で発生する順番で表示される。スクリーン1つについてコマンドを1つだけ選択できる。スクリーンが表示されると、そこにあるすべてのコマンドも表示される。したがって、スクリーン1つにつき最初のコマンドを選ぶことが許される。その場合、他のコマンドは抑止される。

ステップ（STEP）モードが指定されると、システムはデータ・キーのどれかが押されるのを待つ。データ・キーを押すとすぐに、全コマンドがスクリーンにタイプされる。次に、ステップ・モードは別のデータ・キーが押されるのを待つ。第2のデータ・キーが押されると、すぐに次のスクリーンがアクセスされる。

ユーザ（USER）モードが指定された場合、またはどのモードも指定されなかつた場合、システムは、ユーザが正しいコマンドを入力して正しいコマンド・エントリ・キーを押すのを持つ。正しいコマンドを1つ入力するのに、3回のチャンスが与えられる。第6a～第6d図を参照して説明すると、第1のエラーに基づいて、第6a図に示されるようにメッセージが表示される。第2のエラーに基づいて、システムは下正確な文字に対する訂正を行う。ユーザは、第6b図に示される指示された訂正（つまり、無関係の文字の消去）を終えなければならない。第3のエラーでは、第6c図に示されるように適当なエントリがシステムによつて表示されるとともに、次のスクリーンがアクセスされる。ユーザがコマンドに対するすべての必要な訂正をなした後で下正確なアテンション・キーを入力すると、付加的なエラー・メッセージが、例えば6d図に示されるように表示される。

エミュケータ・プログラムで使用可能なキース

特開昭 62-214437 (14)

トロークが下記第7表に示されている。これらのキーの定義および機能は任意であつて、設計者の要求または好みに応じて変化できることを理解すべきである。本実施例では、IBMパーソナルコンピュータ(PC)に使われているキーボードが下で説明されている。これらのキーはすべてコントロール(Ctrl)または代替(Alt)キーと一緒に付けて使われる。すべての定義の形式は、

Ctrl- または Alt- であり、ユーザが Ctrl と他のキー、または Alt と他のキーを同時に押す必要があることを示している。

第7表

キー	ファンクション	説明	キー	ファンクション	説明
CTRL-A	オート・モード	エジュケータをオート・モードにする。	CTRL-PGDN	後続スクリーン	ユーザを次のシステム/ヘルプ・スクリーン・ディスプレイへ前進させる。
CTRL-HOME	パッケージの始まり	ユーザをスクリーン、コマンド、およびヘルプ・ファイルの始まりに置く。	CTRL-N	ノート・ツ・スクリーン・トグル	システム・スクリーンと関連するノート・エリアを該スクリーンの底に表示する。
CTRL-C	コマンド・モード	エジュケータをコマンド・モードにする。	CTRL-PGUP	先行スクリーン	ユーザを先行するシステム/ヘルプ・スクリーン・ディスプレイへ戻す。
CTRL-Q	順会モード	次のノートを求めてスクリーン・ファイルを探索する。	ALT-0	モノ・オンリー	モノクローム・スクリーンだけを使う。
CTRL-S	ステップ・モード	エジュケータをステップ・モードにする。	ALT-1	カラー・オンリー	カラー・スクリーンだけを使う。
CTRL-U	ユーザ・モード	エジュケータをユーザ・モードにする。	ALT-2	モノ・アンド・カラー	モノクロームとカラーの両方のスクリーンを使う。
CTRL-V	ライト・ア・ノート	70文字のノートを書いてスクリーンと共にセーブすることを可能にする。 スクリーンの底の近くにウインドウを表示する。ENTERキーを押すと、直ちにノートはスクリーンと共にセーブされる。	Ctrl-F1 を入力すると現れるエジュケータ・ヘルプ・スクリーンは、下記第8表に示す通りである。なお、各キーに対応するファンクションが英語で示されているが、その訳および意味は上記第7表の通りである。		
CTRL-X	イグジット	PCオペレーティング・システムへ戻る。ユーザがEXITをタイプするとエジュケータへ戻る。	第8表		
ALT-R	タイプ・ライト	普通のタイピングを指定する。	Ctrl-U	USER mode	- interactive learning
ALT-L	タイプ・レフト	キーがタイプされた後でカーソルが左へ動くことを指定する。	Ctrl-S	STEP mode	- semi-automatic sequence of images; any key to move on
ALT-D	タイプ・ダウソ	カーソルが下がる。	Ctrl-A	AUTO mode	- automatic sequence of images with time delay
ALT-U	タイプ・アップ	カーソルが上がる。	Ctrl-G	GOTO mode	- jump directly to specified number screen

特開昭62-214437 (15)

```

Ctrl-L LIST mode - select commands from a list of all
                   that exist
                   - ENTER key begins displaying those
                     selected
Ctrl-N NOTE mode - read any existing note about the
                   system screen
Ctrl-V WRITE mode - write note commenting about a
                   screen
                   - ENTER key writes note to disk
Ctrl-F9 SWAP scrn - single screen systems only;
                   displays help screen

Ctrl-F7/Ctrl-PgUp = Backward Ctrl-F8/Ctrl-PgDn = Forward
Ctrl-HOME = First Screen   Ctrl-END = Last Screen
Ctrl-E/Ctrl-F3 = End Program

Alt-0=Mono Only  Alt-1=Color Only  Alt-2=Both Screens

Press any key to return to system screen

```

教育ソフトウェア・パッケージを教育ドライバ32によって実行するために、ユーザはコマンド・エミュレータとファイルネームを入力する。ファイルネームは、以前に生成された教育パッケージと同じ名前でなければならない。実行例でのファイルネームは I S I L である。

された文字は、実際のホスト・システム上のようにシステム・スクリーン上に現れる。ユーザが該スクリーンに必要とされるコマンドをタイプし、かつ正しいエントリ・キー (ENTER, PFキー、またはCLEAR) を押すと、続くシステム・スクリーンと共に該スクリーン用のインストラクション・テキストも表示される。該システムは I S I L 教育パッケージにおいて编程されてしまっている。

上述のように、テキスト図面 4a ~ 4p は、それぞれスクリーン・テキスト図面 2a ~ 2p に関連している。ユーザはこの機能での操作を経け、必要なインストラクションをタイプする。インストラクションを入力するのに必要なキーストロークは必ずしもコマンドに特有でなくてもよいが、コマンド自身は正しくなければならない。

また、上述のように、不正確なコマンドがスクリーンに入力された場合、ユーザには複数のレベルが与えられる。第 6a ~ 第 6d 図は、与えられた 3 つのエラー・レベルを示す。第 1 のエラー・

本実施例では、本発明を実施するのに 2 つのスクリーンを用いている。第 1 のスクリーンはスクリーン・イメージを順次に表示し、第 2 のスクリーンはスクリーン・イメージと並行して呼び出されるコマンドを表示する。しかしながら、他の実施例では、0 / 1 出力デバイスのパラメータを使って、スクリーンとコマンドの両方を同じターミナルつまりコンソールに表示することができる。

教育ドライバには 2 つの基本モードがある。個人教授・訓練モードと、オンライン・ヘルプまたは参照モードである。2スクリーン・システムでは、システム・スクリーンがカラー・モニタに現れ、ヘルプ・ファイルがモノクロームに現れる。

インストラクション・テキストつまりヘルプ・スクリーンは、システム・スクリーンに関連して編集された情報を表示する。システム・スクリーンに入力する必要のあるコマンドも表示される。実行例において、最初のコマンドは X E D I T I S I L D E M O S C R I P T A 1 である。これがユーザの入力したコマンドである。タイプ

レベルでは、メッセージ (第 6a 図) が、入力の際にエラーが生じたことを示す。同じコマンドについて第 2 のエラーが生じると、足りない、または不正確な文字のどれもがシステムによって訂正され、かつ 1 色で強調される (第 6b 図)。応答に必要とされない文字は、どれも別の色で強調される。ユーザは、関係のない文字の消去を求められる。第 3 の不正確な試行では、システムによって正確な応答がスクリーンにタイプされ (第 6c 図)、スクリーンの底に短いメッセージが表示されるが、そこでは正しい答が表示中であることを示す。なお、以上の説明は、そのまま第 6a ~ 6d 図の英文の大意である。不正確なエントリ・キーを押した場合 (例えば、ENTER が求められているのに、PF5 を押した場合)、適切なメッセージがスクリーンの底に現れ (第 6d 図)。ユーザに対して ENTER の使用を指示する。ユーザは、ソフトウェア・パッケージを通してファイルの終端まで作業を続行し、インストラクションおよびヘルプ・テキストの要求するコマンドをタ

特開昭 62-214437 (16)

イブして、シミュレートされたシステムにおいてシステム・スクリーンを進んで行く。

上述の実行例では、エディタ・プログラムを使って生成されたヘルプ／インストラクション・テキストが第4n回に示されるスクリーンで終了する。該スクリーン（第4n回）で始めると、このスクリーン・イメージがエジュケータ・プログラムによって生成され、基礎インストラクション・テキストと呼ばれる。

オンライン・ヘルプ・モードは、リスト・モード機能を用いると最もよく機能する。この機能は、システム・スクリーンにウインドウを表示する。ウインドウには、この教育パッケージの中で使用可能なすべてのコマンドのリストが表示される。コマンドとコマンドが現れるページ番号の両方が表示される。

該リストは、完全なスクロールおよび選択が可能である。ユーザは、リストをスキヤンし、見たいと思う任意の項目の番号の前にS（「セレクト」を意味する）をタイプし、ENTERキーを押す

オンライン・ヘルプはタスク志向と呼ばれる。そして、ユーザに対して、コマンドを使うスクリーンのタイプ、コマンドの構文、そしてそのようなコマンドを入力して得られそうな結果を示すのに使われる。

教育ドライバのオンライン・ヘルプ・モード

従来のオンライン・ヘルプのほとんどの形式はメニュー駆動式であり、複雑かつあいまいである。ユーザは、ヘルプを探索して得た項目の使い方に因して完全な理解を得ない。構文がしばしば示されるが、実際の使い方はほとんど説示されない。しかしながら、エジュケータ・プログラムは、タスク志向式でオンライン・ヘルプを提供する。これは、指定された項目について、構文と文脈上の使用法を示すことによって達成される。

オンライン・ヘルプ能力を利用する場合、本発明によって設計された教育パッケージはより有効なものになる。このようにして、教育パッケージは教育以上のものになる。ユーザがシステムの専門家であつても、依然有用である。専門家、該パッ

ことができる。ENTERキーを押すと、システム・スクリーンおよびヘルプ／インストラクション・スクリーンが最初に選択された項目に固定する。ユーザが任意のキーを押して自動的にステップ・モードを入力すると、システム・スクリーンにタイプされたコマンドを見、続く任意のキーを押して結果として生じるスクリーンを見ることができる。対応するヘルプ／インストラクション・テキスト・スクリーンが同時に表示される。リストから1つ以上の項目が選択された場合は、再びどれかのキーを押すと、次に選択されたスクリーンがステップ・モードへ移る。ユーザがステップ・モードにとどまつてリストから選ばれた全項目が表示されるまで、このようなことが続く。

この過程の間のいつでも、ユーザは、リスト・モードを再入力して、他のスクリーンを選んだり、以前選択したスクリーンの選択を取り消したり、既に見たスクリーンを見たりすることができる。このような以前に表示されたスクリーンは、アスタリスクを付けて注釈される。このような種類の

パッケージをPCレベルでロードすることができるとともに、トグリングによって、ホスト・システムの問題について参照するために該パッケージをアクセスすることもできる。項目特有ヘルプ・パッケージを、例えば、メニュー駆動式システムのすべてのオプション、あるいは、ワード・プロセッサのタグのすべての動作を説明するように設計することができる。リスト項目としてあげることのできるものは何でも、タスク志向オンライン・ヘルプのサブエントになれる。

エジュケータの中では、リストおよびコマンド・モードが、オンライン・ヘルプについて最も有用である。リスト・モードは、パッケージの中の使用可能なコマンドすべてのウインドウを表示する。これは、スクロール可能かつ選択可能なリストであり、ユーザがヘルプを望む領域について選択をなすことを可能にする。

コマンド・モードは、リストが示されない点を除くと、リスト・モードと同じである。リストが示されない代りに、ユーザが情報を望む文字列を

特開昭62-214437 (17)

タイプすると、リストが探索される。コマンド・モードで実行される探索では、まずユーザ・エントリとリスト上のコマンドとの間の完全な突合せ（マッチ）の配置が試みられ、次にサブストリング・マッチの配置が試みられる。効果は、リスト・モードの場合と同じである。ユーザには、要求したものについてのヘルプが示される。リストの中のコマンドは、入力タイプ・キーによって分離される。すなわち、選択可能なコマンドのそれぞれの前にはエントリ・キーが先行し、同じく後にはエントリ・キーが続く。多くの項目がホスト・システム・スクリーンに現れるかもしれないが、リスト・モードの項目の1つだけが選択可能である。これは、スクリーンから項目を1つ選ぶと、他のすべての項目も表示されるという理由による。

教育モードの生成には、代替例もある。それによると、以下のステップが含まれる。

a) 上述のように、SESSIONがスクリーンとコマンドを捕獲するのに代って、エディタを用いてこれらを生成し、

イピングの理論に關係する。本発明のシステムによれば、視覚的に意味のあるスクリーンとコマンドの両方の高速プロトタイピングが可能になるとともに、機能を示すことも可能である。プロトタイプが完成すると、直ちに任意の数の人間に送ることができる。そして、該プロジェクトについて膨大なコーディング時間を費す前に、これらの人間からコメントを収めることができる。プロトタイプのレビュー・承認が終るとすぐに、後でコンバート・プログラムに関して説明するように、既にPCレベルで設計されたスクリーンを適切な言語に変換してホストにアップロードすることができ、その結果、作業を始めからホスト・システム上に置換する必要がなくなる。スクリーン・デザイン自身も、このようにしてより一層シンプルにできる。完全なスクリーン・エディタによって、ほとんどのシステム・ツールと他のプロト・タイプのデザイン時間を要する色彩定義およびフィールド定義文字をバイパスして、プロトタイプの生成はなされる。

b) 下記のようにコンバートを用いてテキスト・ファイルからコマンド・ファイルを生成するとともにSESSIONを用いて該コマンドをプレイバックし、関連するイメージ・シーケンスを自動的に挿入。

c) 任意のエディタを用いてイメージおよびGM・L様タグのセットを使うコマンドのテキスト・ファイルを生成し、コンバートを用いてエミュレータ・プログラムが使うスクリーンおよびコマンドのファイルを生成する。

試験モード

試験は、繰り返しの、または同様のコマンド・セットを使う場合でも、実行の度に違うデータを出すと期待される項目に関連する。そのような試験の2つのカテゴリーは、ヒューマン・ファクター試験および性能試験である。ヒューマン・ファクター試験は、基本または拡張のどちらかに分類することができる。

基本ヒューマン・ファクター試験

基本ヒューマン・ファクター試験は、プロトタ

拡張ヒューマン・ファクター試験

ラボラトリ環境での試験、計時データの収集、およびユーザが行って見るものとの理解は、すべて拡張ヒューマン・ファクター試験の一部である。これは、製品がどの程度役に立つかを決定するのに実行される試験である。本発明のシステムは、次のような能力を持つプログラムを含む。

- a) ユーザが1つのスクリーンから別のスクリーンへ進むのに要する時間（システム・レイアウトを含む場合と含まない場合がある）を測定する。
- b) 各スクリーンでユーザが入力したキーストロークを正確に示す。
- c) いつでも他人が見たりレビューしたりできるようにユーザの経験全体を再生成する。

この結果、ストップウォッチ、オーディオ・テープ・レコード、ビデオ・カメラのような、従来のヒューマン・ファクター測定・観察器具が、かつてのように置換できなかつた状態を脱した。今ではこれらを全部取り除いてもよい。さらに、ラボラトリでもはや試験を行う必要がない。試験は

特開昭62-214437 (18)

サブジェクト自身の環境で行える。これは、各ユーザがPCつまり同様のスマート・ターミナルを持ちさえすれば、従来よりもはるかに多数の視聴者を相手にできることを意味する。

性能チャートは、第9表のようなフォーマットを持つ。

第9表

経過時間	中の時間	タイムスタンプ
command : <コマンドのテキスト>		
>send :		
<back :		
command : <コマンドのテキスト>		
>send :		
<back :		

横の第1列は、時間0:00:00.00に始動し、コマンド毎に増加する。第2列は、コマンド毎に時間0:00:00.00に始動し、ホストが入力に対応できるようになると直ちにPCの応答時間を与え(>send :)、PCの入力が完了すると直ちに、ホストの応答時間を与える(<back :)。第3列は、コマンドが発せられてその後の応答があつた日の時間を印刷する。

横の第1行は、コマンドを表示する。第2行は、PCまたはユーザがコマンドを発するのに要するタイム・ビリオドを表示する。該時間は、ホスト・システムが入力に対応できるようになる時点から測定され、終了時間はユーザがエントリ・キーを押す時である。第3行は、システムが応答して付加的な入力に対する対応準備が整うまでに要するタイム・ビリオドを表示する。始動時間は、ユーザがエントリ・キーをタイプする時点から測定される。終了時間は、ホストが付加的入力を受け取るようになつた時である。

性能チャートの終わりでは、印刷された要約が、

発せられたコマンドの数、平均(>send : and <back :)時間、最長(>send : and <back :)時間、最短(>send : and <back :)時間、および合計(>send : and <back :)時間を表示する。
性能試験

システム上のユーザのターミナルにおいてシステム応答時間を判断することは、しばしば困難な仕事である。通常は、試験を正しく行うために、ストップウォッチを持つた人間がターミナル・オペレータを観察するとともに、コマンドに因連する各システム応答の時間を計る。しかしながら、これでは時間がかかり、かつ不正確である。ユーザに因連する計時情報を収集することも困難である。

SESSIONを・P(性能チャート)オプションで用いることにより、タイマや特別のキーストロークでクロックを始動・停止する必要なしに、このような情報を集め、かつセーブすることができる。性能オプションによれば、最初のシステム・アテンション・キー(ENTER, CLEAR,

任意のPFまたはPA)から計時が開始される。これらのキーの1つを押すとタイミングが取られるとともに、システムが、「使用可能」「入力は禁止されていない」というメッセージを戻すと、別のタイミングが取られる。この時点で、ユーザによるデータの入力が可能になる。真のエンド・ユーザ応答時間を決定することができる。ユーザがデータを入力する際、クロックは別のシステム・アテンション・キーが入力されるまで走り続ける。別のシステム・アテンション・キーが入力されるとき、計時が再び始まる。そして、ユーザ間は・Pファイルにセーブされる。しかしながら、PCがコマンドをシステムに自動的に従わせること、およびユーザの介入なしでシステム応答時間の統計を集めることが好ましい。これを行う手段は、後で再試験に因連して記されている。

・Pファイルは、計時(性能)情報のすべてを保持している。該ファイルの終端における要約は、コマンドの数、最大のシステム・ラグ時間とユーザ・エントリ時間、最小のシステム・ラグ時間と

特開昭62-214437 (19)

ユーザ・エントリ時間、合計のシステム応答時間とユーザ入力時間、および平均のシステム応答時間とエンド・ユーザ入力時間とを含む。個々のコマンドは行にリストされ、計時情報は列にリストされる。

システム性能テストが生成する性能チャートは、ヒューマン・ファクター・テストが生成するものと同じである。さらに、性能テストと性能チャートに現れ、計的のトリガとなる事象は、ヒューマン・ファクター試験に因縁する前述の事象と同じである。

再試験モード

再試験は、以前にテストした製品の試験を含む。製品が一度試験されている場合は、該試験のある程度の記憶の存在が期待される。そのような記憶は、PCの記憶であつてよい。このように、PCは以前実行した試験を戻し、新しく得る結果を古い結果と比較することができる。また、PCを用いて、発見された問題をその発生場所も含めてドキュメントしたり、因縁情報を繰り返し表示した

このような初期試験の際にSESSIONプログラムを用いたならば、自動試験ケースが既にしきいテストについて存在している。

SESSIONのプレイバック・ファイル・オプション、および・Dパラメータは、再試験に因縁している。プレイバック・オプションは、人間オペレータがPCキーボードについているかのように、SESSIONがホスト・システムを駆動することを可能にする。しかしながら、PCがシステムを駆動する場合、オペレータのミスや遅れは存在しない。PCは、ホストが許す限りいくらでも速く実行する。

ホストをプレイバックするのに用いるファイルは、以前のリリースの試験を以前に記録したものである。これが、テスト・ケースを意味する。

・Dパラメータ（差異ファイル）パラメータを用いることにより、SESSIONは、コマンドをプレイバックするだけでなく、ファイル上のスクリーンに関して以前実行したものと現在プレイバックされて表示中のものとの比較も行う。発見さ

りできる。PCを旧リリースに対する新リリースのしきい試験モードで使うにせよ、性能しきいを越えた直後の製品のインライン・テストで使うにせよ、または、リリースされてサポートが必要になつた直後の製品のフィールド・エラーの通报に使うにせよ、本発明のシステムは、人間試験者よりもこれらの試験をより迅速にかつ能率よく指揮することができる。

しきい（回帰）試験

リリースを重ねる度に、ソフトウェア製品は変化し、かつ高級になる。しかしながら、このような高級化においても、オリジナルの機能を依然としてサポートするとともに、試験を施しておくべきである。これは、しばしば回帰試験と呼ばれるものであり、新しい機能が試験された後で生じる。SESSIONを使うと、回帰試験は、製品が人間試験を受け取るよりも十分前に存在しなければならない最小限の受容可能機能のしきい試験となり得る。機能の最小限セットは、初期のリリースにつき以前に実行した試験によって決定される。

された差異はどれも差異ファイルにセーブされ、後で人間がレビューする。セーブされるのは差異だけである。スクリーンの数に差がないならば、SESSIONは、単にコマンドをホストに出し、かつ差異を識別し続けるだけである。

実行例を用いると、次のようにSESSIONを呼び出してしきい試験を実行することが可能である。

SESSION ISILD PLAYBACK

SESSIONは、該例について以前に記録されたコマンドを自動的に検索し、どんな差異も検出する。比較のために使うスクリーン・ファイルは、第2a～2p図に示される編集前のシーケンス・イメージ・ファイルである。

第7図には、再試験に用いるシステム構成が示されている。しきい試験が完了すると、直ちにレビュー・プログラム50による差異ファイル26のスキヤンが可能になる。イメージ・ファイル26とコマンド・ファイル28もレビュー・プログラム50へ入力される。このプログラム50は、

特開昭62-214437 (20)

以前のランからオリジナルのスクリーンを示し、続いてそれに現在のラインの際に見つかった差異を並ねる。このような差異は強調すると、単一の文字の場合であつても認識が容易になる。特別情報は、レビュー・スクリーンの底へ出てくる。そのような情報は、該スクリーンに到達するのに用いたコマンド、オリジナル・スクリーンのファイルの中のスクリーン番号、および差異を有するスクリーンの合計数を含む。

レビュー・モード

レビュー・プログラムによると、ユーザが、SESSIONにおいて・Dオプションを持つファイルをコンソールがプレイバックした後、コンソールが扱ったデータを見れるようになる。オリジナルのユーザ・セッションが表示され、かつPCが発見したセッションにおける差異がオリジナルに重ねられる。ユーザは、オリジナル・セッションとPCラン・セッションの間で切り替えて、2つのランにおいて異なるスクリーンだけを見ることができる。

に依存している。レビューにおけるオリジナル・スクリーンから異なるスクリーンへの切替は、まづ比較的遅く生じる。これが直接的な全後続時間である。この結果、大きな差異のあるスクリーン上で、差異が突然現れることはないが、スクリーン上への表示には色が使われる。

レビュー・コマンドは次のような構文を持つ。
REVIEW <ファイルネーム> [・レベル] [アスタリスク] [分離記号]
レベルは、DOSにおけるファイル・エクステンションと同様のものである。これによれば、使用中のファイルのデフォルトのレベルが指定されたレベルにセットされる。使用されないときのデフォルト・レベルは、そのファイルのタイプ(.C, .S, および.D)についての存在レベルの最高値となる。指定レベルは1から99の間の数であつてよい。アスタリスク・パラメータは、分離スペースなしで直にファイルネームに続かねばならない。分離パラメータはファイルネームから分離されなければならず、かつ互いに少なくとも1スペース離れていなければならない。

レビューによると、ユーザが保護(非入力)フィールドと無保護(入力)フィールドの差異だけを見ることが可能になる。このようにして、異なるシステムのある特性をマスクし、あり得る差異を無視することができる。エディタによつても、オリジナルのユーザが捕獲スクリーン上のマスク文字の生成が可能になる。SESSIONはこれらのマスク文字を無視するので、かかる物理的なスクリーン位置で生じるどんな差異も続くプレイバック時には無視される。

レビュー・プログラム50はターミナル34に常駐し、ターミナル34によつて呼び出すことができる。すべてSESSIONプログラム12(第1図)によつて生成されるイメージ・ファイル26、コマンド・ファイル28、および差異ファイル22は、別々にレビュー・プログラム50に入力できるし、どのように組み合わせて処理してもよい。

レビューは、・Dオプションを使ってプレイバック・モードで実行されつつあるSESSION

2スクリーンPCを使う場合、カラー・オンリーは、レビューをカラー・スクリーンへ強制的に向ける。

例： REVIEW testfile*1

2スクリーン・システムが使用中ならば、システム・スクリーンをどちらかのディスプレイに配置することが時には有用である。*1のために、レビュー・プログラムはカラー・スクリーンだけしか使うことができない。モノクローム・ディスプレイは、自由な状態のままである。

2スクリーン・コンソール・システムでの場合に、モノクローム・オンリーは、レビューの出力をモノクローム・スクリーンへ強制的に向ける。

例： REVIEW testfile*0

コマンド・ファイルを確立するとときは、ユーザ指定コマンド・ファイル・エクステンション番号が用いられる。

例： REVIEW testfile c5

このパラメータはレベル・パラメータのバリエーションであり、コマンド・ファイルだけについ

特開昭62-214437(21)

てファイル・レベルの指定を許す。

ユーザ指定差異ファイル・エクステンション番号は、差異ファイルを生成するのに用いられる。

例: REVIEW testfile b5

このパラメータも、レベル・パラメータのバリエーションであり、差異ファイルだけについてファイル・レベルの指定を許す。

保護オブリー機能は、文字Pによって呼び出される。レビューは、保護フィールドで見つかった差異だけを表示する。

例: REVIEW testfile p

保護フィールドで見つかった差異も表示される。他のすべて（無保護フィールド）は、差異として表示されない。

差異(.D)ファイルを生成中の間、このオプションでSESSIONを実行することも可能である。その場合、差異ファイルは保護フィールドの差異だけを備える。そうでない場合、SESSIONはすべての差異をセーブする。そして、それらを一括するには、レビューを呼び出さねばならない。

その場合、差異ファイルは無保護フィールドの差異だけを備える。そうでない場合、SESSIONはすべての差異をセーブする。そして、それらを一括するには、レビューを呼び出さねばならない。

ユーザ指定スクリーン・ファイル・エクステンション番号は、スクリーン・ファイルの選択に使われる。

例: REVIEW testfile a5

このパラメータも、レベル・パラメータのバリエーションであり、スクリーン・ファイルだけについてファイル・レベルの指定を許す。

実施例では、下記第10表のようなコントロール・キーの組合せが最も有用であることがわかつている。

第10表

キー	ファンクション	説明
CTRL-HOME	ファイルの始まり	ユーザをファイルの中の差異の始まりに移す。

ない。

クワイエット機能は、プログラムの事前、事後のログおよび音声のほとんどをオフにする。

例: REVIEW testfile q

プログラム・スタートおよびエンドのどのログも、ログのタイポマチック・サウンドとともに抑止される。qがプログラムのファイルネーム・メニューから指定されたならば、プログラム・エンディング・ログだけが抑止される。これに対して、イニシャル・プログラム・ログは既に表示済である。

無保護オブリー機能は、文字Uによって呼び出される。レビューは、無保護フィールドで見つかった差異だけを表示する。

例: REVIEW testfile p

無保護フィールドで見つかった差異はどれも表示される。他のすべて（保護フィールド）は差異として表示されない。

差異(.D)ファイルを生成中の間、このオプションでSESSIONを実行することも可能であ

CTRL-E F3	エンド・レビュー	F3レビュー・プログラムを終了させ、DOSへ戻る。
CTRL-END	エンド・オブ・ファイル	ユーザをすべてのファイルの終端へ移す。
CTRL-PGDN CTRL-F8	後続スクリーン	ユーザを後続の差異ディスプレイへ移す。
CTRL-PGUP CTRL-F7	先行スクリーン	ユーザを先行する差異ディスプレイへ移す。

試験が一旦レビューされると、人間試験者は見つかった差異はエラーであるかについての判断をなすことが可能になる。

自動試験

自動試験は、しきい試験が完了した後に行われる試験を扱う。開発されたコードが、新しい機能を実行する責任を持つ人間試験者の手にある。人間試験者がSESSIONの下で操作しており、かつ少なくともコマンドをセーブしている限り、ジョブをより一層容易にすることができます。

SESSIONによればコマンドとスクリーンのセーブが可能になり、その結果、人間試験者が

特開昭62-214437 (22)

簡単にステップ列全体をリプレイすることができます。試験者がエラーを見出し、該エラーが開発者によって訂正されると、直ちにコードを再試験して修理を保証しなければならない。この再試験では、元のエラーの原因となつたステップを正確に繰り返すべきである。このようなコマンド列はしばしば覚えにくく、繰り返すのは時間の浪費である。人間試験者が元のエラーへ至るパスで他の何らかの変化が生じたことを知ることも困難である。

SESSIONは、元のエラーへ至るキーストローク列を正確にリプレイするとともに、しきい試験の場合のように実行された場合は、試験において該時点までに発達した差異にも吾及する。自動試験では、SESSIONを用いて試験者の骨折り仕事を使らすことができる。

コンパート・モード

コンパートは、SESSIONによって生成されたファイルを人間の読解可能な形式に変更する翻訳プログラムである。コンパートは、人間が読めるファイルをSESSIONが読める形式に変

プログラムを使ってコマンド・ファイルを記録する。ユーザは、SESSIONの中のタスク開始定義コマンド (Ctrl-M) を出す。すると、対話的なタスクが普通に行われる。該タスクが完了すると、タスク終了定義コマンド (これも Ctrl-M) が入力され、最初に指定されていたタスクがコマンド・ファイルとしてセーブされる。コマンドを生成する第2の方法は、単にエディタを使い、以前定義したようなコマンドのリストを生成することである。続いてこのリストはコンパート・プログラムによって処理され、実行可能なコマンド・ファイルになる。

SESSIONフォーマットに変換し直されたファイル42は、韓文とロジックがチェック済である。これは、ベーシックやバスカルのプログラムをコンパイルするのに似ている。ここでの文脈において、コンパイルによる類推は、SESSION読解可能コマンドと人間読解可能コマンドの両方を備えるファイルに關係する。エラーにはフラグが立てられ、重大性に応じて、変換は続

更することができる。

図8図には、イメージ・ファイル26、コマンド・ファイル28、およびインストラクション・ファイル30が接続されたコンパート機能40が示されている。ファイル26、28、30はまとめて、セーブされたセッション・ファイルと呼ばれ、参照番号42が付されている。3個のテキスト・ファイル46とホスト・フォーマット・イメージ出力ファイル48は、一括して参照番号44が付された人間読解可能ファイルを形成する。

人間読解可能ファイル46は、PCエディタを使って編集することができる。コマンド言語を使うと、新しいコマンドを挿入したり、古いコマンドを削除したり、ファイルを拡大・縮小したり、ファイルを使って全く新しいファイルを生成したりすることができる。SESSIONが該ファイルを使用できるようにするには、コンパート40をもう一度実行しなければならない。

コマンド・ファイルは、2つの方法のどちらかで生成できる。第1の方法では、SESSION

行または終了される。ファイルはSESSIONフォーマットに直され、存在エクステンション番号 (つまり、.Cナンバー、.Sナンバー、または.Hナンバー) の最高値に1を加えた値が与えられる。したがつて、ファイルに重ね書きが行われることはない。

コマンド・ファイルに対してコンパートを使う利点の1つとして、コンパートのサポートする言語は、SESSIONが処理できるエミュレータ・キーを1つ以上許容することが挙げられる。これによれば、コマンド・ファイルが元々そう作られているのよりも一層一般化されたタスクを達成することが可能になる。コンパイルの1つのタイプがファイルについてなされるので、スクリーン探索、セーブ、および可変パラメータと同様に、ルーピング制御がサポートされる。これらの言語コマンドのあるものはSESSIONに限られているキーをシミュレートする。その他の言語コマンドは該言語だけに限られている。SESSIONがレコーディング中に特殊言語制御を捕えること

特開昭62-214437 (23)

はできないけれども、コマンド・ファイルの形になつた制御を理解することはできる。

人間読解可能ファイルを SESSION 読解可能にすることに実行するコンパイルは、補正（コレクティブ）コンパイルの 1 タイプである。問題が修正点を過ぎてコンパイルされるか、または問題があまり難しいときだけ、コンパートはコンパイルを中止する。しかし、エラーがコンパイルの際に訂正可能だとわかると、該エラーは訂正される。

必要なコマンドが落ちた場合、可能ならば、コンパートはそのようなコマンドを挿入し、SESSION の要求を満たす。このような付加されたコマンドは、元々人間読解可能形式から変換された SESSION 読解可能ファイルを再び人間読解可能形式に変換し直すときに目立つ。このような操作は、コマンド言語ファイルをデバッグするのに有用である。なぜなら、コンパイラの変更するコマンドが明白になるからである。このような二度の変換を行つたファイルを注意して検査する

コンパートを用いて、コマンド・ファイルを生成したり、修正したり、一般化したりできるし、あるいは反復する自分を備えたホスト・セッションにすることもできる。

エラー通報

エラー通報は、可能ならばコード開発者が問題を考えたり除去したりできるように、試験者およびユーザが問題をドキュメントする処理である。従来の文書化方法は、しばしばシステムをまずく表現していた。この結果、訂正が発効するまでの長い遅延、何が問題であるかについての混乱、および問題を決定する試みにおいての文書化されたエラーを越えて特別に通信を行う必要が生じる。さらに、ユーザは、エラー発見の際に使うステップ列を正確に覚えてなくてよいから、エラーはしばしば正確に文書化され、開発者には無効であると思われてしまう。

SESSION を使うことにより、ターミナル・セッションについて正確なレコードが保持される。このレコードはコマンドを含むだけではない。ユ

こと、および三度目の変化は行わないことが認められる。

操作時に、コンパート・プログラム 40 は、コマンド・ファイル 28、イメージ・ファイル 26、インストラクション・ファイル 30、またはこれらの任意の組合せをテキスト・ファイル 46 に変換できる。さらに、イメージ・ファイル 26 は、コンパート・プログラム 40 によってホスト・フォーマット・イメージ・ファイル 48 に変換可能である。

コンパート・プログラム 40 を呼び出すには、次のような構文を使わなければならない。

CONVERT <ファイルネーム> [fromc|toc] [cf] [nonus]

ここで、ブラケット [] でオプショナルであることが示されている前述のパラメータは、下記第 1 表の機能を表わしている。

第 1 表

fromc toc	変換の向き
cf	変換するファイルのレベル
nonus	カーネル・ロケーションを使わない

ユーザが問題に至るアクションをさらに記述するために同時に付加することのできるテキストのただ 1 行だけでなく、ユーザが見るスクリーン全体も含む。これは前記のノート・ツ・スクリーン (NOTE TO SCREEN) によって達成される。エディタ・プログラムを使うセッションが終了した後、ユーザ指定テキストのスクリーンまたはページの全体を入力できる。このような文書化によって、開発者は、ユーザが入力したコマンドだけでなく、ユーザがそれらを入力した時間、そしてその時にユーザが何を見て何を覚えていたかを見る。

再構成可能自動タスキング・システム

エミュレータ・オーバーレイとしての SESSION には、普通のエミュレータだけでは得られない機能がある。マクロは、PC レベルでの BAT ファイル、またはホスト・システムでの CLISTS もしくは EXECS に似ている。SESSION を使って単にコマンド列を記録するだけで、マクロを生成することができる。このような

特開昭 62-214437 (24)

マクロはいつでもブレイ・パンクでき、マクロ・ファイル・ネームに至る反復的で複雑なキーストロークを繰り返す。

本実施例では、下記第12表のコマンドがキーボードによって挿入され得るすべてのコマンドの他に、後記のようにマクロをもつと柔軟つまり再構成可能にすることができる付加的なコマンドも含むセットを形成する。

第12表

キーボード・ファンクション

[INSOF]
[INSON]
[PRTSC]
[ENTER]
[TR]
[EEOI]
[EOF]
[HOME]
[B]
[TAB]
[BTAB]
[U]
[D]
[CR]
[R]
[L]

[END]
[PF1]
[PF2]
[PF3]
[PF4]
[PF5]
[PF6]
[PF7]
[PF8]
[PF9]
[PF10]
[PF11]
[PF12]
[PA1]
[PA2]
[CLEAR]
[DEL]
[RESET]
[BEEP]

可変パラメータ

[x0]
[x1]
[x2]
[x3]
[x4]
[x5]
[x6]
[x7]
[x8]
[x9]

タイム・スタンプ

[T]
ストリングの区切りを見つける

[F+]
[F-]
見つかられば実行する

[+F]
見つからなければ実行する

[-F]
CTRL-#まで待つ

[WAIT]

キーが押されるまで休止する

[PAUSE]

開始と終了

[+]
[-]

コマンド・ファイル・ネームの区切り

[C+]
[C-]

入力を繰り返すことによるSESSIONの進行の停止および保持

[HOLD]
[STOP]

GOTOのためのラベル

[:0]
[:1]
[:2]
[:3]
[:4]
[:5]
[:6]
[:7]
[:8]
[:9]

GOTO コマンド

[G0]
[G1]
[G2]
[G3]
[G4]
[G5]
[G6]
[G7]
[G8]
[G9]

ノード・オペレーション

[NOP]

近延指定タイム・ピリオド

[DELAY]

特開昭 62-214437 (25)

<u>使用システムの特別チェック</u>	[R5] [R6] [R7] [R8] [R9]
[VM] [NVS]	
<u>パスワードの区切り</u>	[READ]
[L+] [L-]	<u>局所変数</u>
<u>スクリーンのイネーブル/ディスエーブル</u>	[VAR0] [VAR1] [VAR2] [VAR3] [VAR4] [VARS] [VAR6] [VAR7] [VAR8] [VAR9]
[BLNK+] [BLNK-]	
<u>特殊カーソル・ロケーション</u>	<u>大文字/小文字</u>
[GOXY] [SNAP] [OLDXY] [(-)] [(+)] [TYPER] [TYPEL] [TYPEU] [TYPED]	[CAPON] [CAPOF]
<u>リセット GOTO</u>	<u>値を変数に割り当てる</u>
[R0] [R1] [R2] [R3] [R4]	[SET0+] [SET1+] [SET2+] [SET3+] [SET4+] [SET5+]
[SET6+] [SET7+] [SET8+] [SET9+] [SET-]	
<u>局所変数の先頭と終の少なくとも一方の消去</u>	きる。
[STRIP]	下記第 1 3 表のマクロ・ファイルは、付加的なコマンドが挿入された結果、ファイルが一連のホスト・ファイル全体を更新することのできる簡単なマクロの 1 例である。FIND、GOTO およびルーピング・コマンドのために、システム状態の変化に適切に反応できるという点で、該マクロは通常のマクロよりも柔軟なものになっている。
コンバートおよびコマンド言語を使って、記録する代りにマクロを書込んだり、記録後に変更したりすることができる。コマンド言語によれば、条件作法、ループ、ディスクへの書き込み、可変パラメータ、および単に記録するだけでは複雑できない他の多くの機能が許される。したがつて、コマンド言語によれば、無限に処理する能力のある強力なマクロを生成することができる。一旦生成または修正された後、統いて複数可能な形のマクロ (.CMD ファイル) がコンバート・プログラムの下で実行される。このため、ファイルは S E S S I O N 故障可能形式 (.C# ファイル) になる。該ファイルは、最も複雑なマクロさえも正しくなるまでコンバートの下で何度でも実行することができる。	このファイルは、一連のファイルを更新するために、X E D I T を (VM 上で) 使うように設計されている。

特開昭62-214437(26)

卷之三

エミュレーション

コンバートを用いると、マクロ・ファイルの一般化、長時間実行タスクまたは反復タスクの生成、毎日の操作の簡略化、およびユーザがホスト・システムと会話をするやり方のカスタマイズが可能になる。データ・セット・ネームや、後で可変パラメータに変換されるような特定入力で、マクロ・ファイルを描えることができる。長時間実行タスクおよび反復タスクは、数個のキーストロークを入力し、ルーピングまたはプランチング・マクロでもつて実行することができる。最後に、コンバートを用いると、ホストを登場させてコマンドとともにパラメータを受諾したり、編集が完了するとデータ・セットを自動的にセーブすることが可能である。例えば、データ・セットを編集するとともに、編集が完了するとそれをセーブするコマンド・ファイルは、下記第14表のように現れる。

第14表

[CLEAR]
EDIT [S0] [ENTER]
[WAIT]
[HOME] SAVE [ENTER]
[HOME] QUIT [ENTER]

ホストのエディタを使うためのキーストローク
は次のようになる。

Ctrl-P

MY-DATASETは[0%]位置を占めており、システムは待機状態のエディタになつてゐる。編集が完了すると、ユーザはCtrl-Wを押して該ファイルをセーブし、エディタを出ることができる。

E. 発明の効果

本発明のタスキング・システムによれば、変化するホスト・システム環境に応答して様々なタスクを自動的に反復して規定、実行することができる。

4. 図面の簡単な説明

第1図は、本発明のSESSIONプログラムによる要素、および該要素にさらに付随的な要素も含めたものの間のデータの流れを示すプロツク図である。

第2a～2p図は、実行例においてコンソール上に表示されるイメージを表わす図である。

第3回は、付随的要素を備えたエディタの構成を示すブロック図である。

第4a～4p図は、それぞれ第2a～2p図に
関連するインストラクション・テキストを表わす
図である。

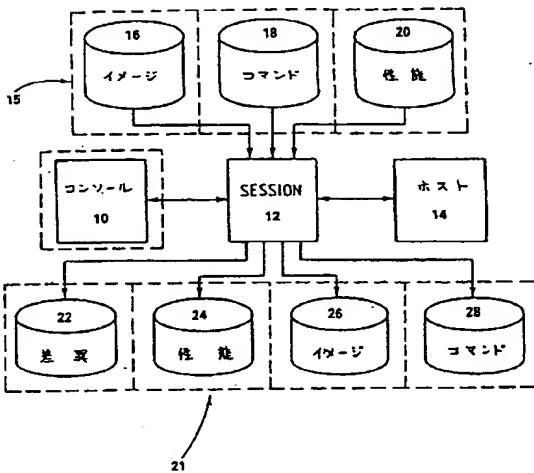
第5図は、教育ドライバおよびそれを用いて操作できるモードのブロック図である。

第6a～6d図は、教育ドライバに関するエラーを表わす図である。

第7回は、本発明による再試験のための構成を示すブロック図である。

第8図は、本発明によるコンパート機能を示すブロック図である。

特開昭62-214437 (27)



第1図

VM READ VM6
1.....2.....3.....4.....5.....6.....7.....8
CNS

四
三
二
一

```
ISRDENO SCRIPT A1 V 132 TRUNC=132 SIZE=0 COL=1 LINE=0  
*****TOP OF FILE*****  
1.....2.....3.....4.....5.....6.....7.....8.....9.....  
*****END OF FILE*****  
*****END*****
```

图 2-6

```
ISOLDRK SCRIPT A1 V 132 TRUNC=132 SIZE=9 LINE=0 COL=1 ALT=0
INPUT MODE:
***** TOP OF FILE ***** 1.....2.....3.....4.....5.....6.....7.....8....9...
***** INPUT ZONE ****
***** INPUT MODE 1 FILE ***** 1.....2.....3.....4.....5.....6.....7.....8....9...
```

22

2 d

Paragraph versus Res
This

—223—

特開昭62-214437 (28)

四
四二

卷之二

四二

第21回

Paragraph and 5	7
This	6
HEAD LEVEL ONE	6
HEAD LEVEL TWO	5
Head Level Three	4
Head Level Four	3
Head Level Five	2
Head Level Six	1

四二一

三

特開昭62-214437 (29)

四二

四二〇

卷二〇

圖 2 m

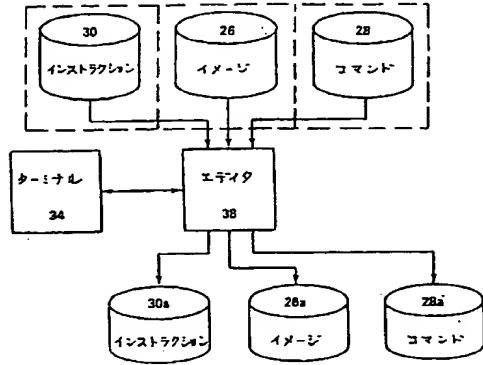
HEAD LEVEL ONE1.....
Head Level Two2.....
Head Level Three3.....
Head Level Four4.....
Head Level Five5.....
Head Level Six6.....
Item 17.....
Item 28.....

```

ISLDEMO SCRIPT A1 V 132 TRUNC=132 SIZE=17 LINE=17 COL=1 ALT=0
XEDIT:
===== 16. Head level six
===== 15. Item1
===== 14. Item2
===== 13. Item1
===== 12. Item1
===== 11. Item1
===== 10. stuff for 1
===== 9. Item2
===== 8. stuff
===== 7. END OF FILE
===== 6. END OF FILE
===== 5. END OF FILE
===== 4. END OF FILE
===== 3. END OF FILE
===== 2. END OF FILE
===== 1. END OF FILE
XEDIT 1 FILE

```

第2p 図



第3 図

Keytitles entered to get here:

This is a demonstration of the ISL GML language, and some simple things to do to get started. To use ISL (or GML) you must first have familiarity with an editor, and be a data-savant to do. Create a data set named ISLDEMO_SCRIPT on your 'A' disk, so that we can start producing some basic ISL tags.

Keystrokes needed for next screen:

edit ISL demo script & ENTER

ISLDEMO SCRIPT A1 Page 1

Keystrokes entered to get here:

This is the standard XEDIT screen, enter input mode so that the ISL tags and text can be used.

Keystrokes needed for next screen:

Input[ENTER]

ISLDEMO SCRIPT A1 Page 2

第4 b 図

Keytitles entered to get here:

Input[ENTER]

↑ - This designates the beginning of a new paragraph. It may be used any time that a new paragraph is to be started. All text following it will be part of that new paragraph.

Try the ↑ command by typing the following test:

↑:Paragraph
use the ↑. This

ISLDEMO SCRIPT A1 Page 3

Keytitles entered to get here:

↑:Paragraph[CR and End ↑ The ENTER]

To see what ISL would do with this type of 'tag', simply press ENTER.

[For test 'isolating' of the ISL tags, you would have to leave the editor and issue a mode command, but this will be covered later. Suffice now simply to demonstrate the effects of the ISL tags on this test.]

Keystrokes needed for next screen:

ENTER

ISLDEMO SCRIPT A1 Page 4

第4 c 図

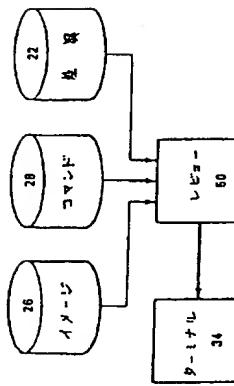
特開昭62-214437 (31)

Keystrokes entered to get here:
[ISL1: Head level one(CR)ht,Head level six(ENTER)]
Again, to see what [ISL1] script would look like, just press ENTER.
Keystrokes needed for next screen:
[ENTER]

第4 i 図

Keystrokes entered to get here:
[ENTER]
Although difficult to see on the screen, a level one head causes a new page to be printed with the header at the top. A level two head is printed in bold-double strike. A level three head is printed in italic double strike. A level four head is printed in bold. A level five head is printed in italic, bold, with the text, and a level six is printed half, text size, italic with the text.
To return to the editor and see the [ISL1] tags for lists, simply press ENTER.
Keystrokes needed for next screen:
[ENTER]

第4 j 図



第7 図

Keystrokes entered to get here:
There are three kinds of lists in [ISL1]:
Simple lists
Unordered lists
Ordered lists
Simple lists are as they seem, simple. Type the following and you'll see what I mean:
1. item1
2. item2
3. item3
...and press ENTER

第4 k 図

Keystrokes entered to get here:
[ENTER]
As you can see, the format of the text file used in XCATF has little bearing on the format of the scripted [ISL1] file.
The [p] tag causes a separation within the text, and a line skip.
To re-enter the editor and see other [ISL1] tags, just press ENTER.
Keystrokes needed for next screen:
[ENTER]

第4 e 図

Keystrokes entered to get here:
[ENTER]
Not we will cover headers. Headers are used to set apart logical segments of similar text. [ISL1] allows 8 levels of headers to be used. The tag for header is similar to that of a paragraph [p]. Although the header tag is followed by a number designating its level.
Type the following to become familiar with the use of [ISL1] headers:
Keystrokes needed for next screen:
[ENTER]

[ISL1 : Page 5]

[ISL1 : Page 6]

[ISL1 : Page 7]

[ISL1 : Page 8]

[ISL1 : Page 9]

[ISL1 : Page 10]

[ISL1 : Page 11]

第4 f 図

Keystrokes entered to get here:
[ht,Head level one(CR)ht,Head level two(ENTER)]
Now type the next two levels as follows:
ht,Head level three
ht,Head level four
...and press ENTER

第4 g 図

Keystrokes entered to get here:
ht,Head level one(CR)ht,Head level four(ENTER)
Finally, type the last two levels of headers as follows:
ht,Head level five
ht,Head level six
...and press ENTER

[ISL1 : Page 8]

[ISL1 : Page 9]

[ISL1 : Page 10]

[ISL1 : Page 11]

四〇四

Keystrokes used to get here:
[ENTER]

Keystrokes used to continue:
:ol.[CR]:.item1[CR]:.item2[CR]

第4章

Keystrokes used to get here:
:!|[CR]:!|item1|[CR]:!|item2|[CR]:!|end|[ENTER]
Keystrokes used to continue:
|[ENTER]

Keycode is needed to get characters
[alt][ctrl][shift][ctrl][alt][ctrl][shift].
1st - displays a simple list
2nd - displays a list item
3rd - displays the end of the simple list
To see what list would do with this, just press **ENTER**
Keycode is needed for next screen!
[alt][enter]

Keystrokes entered to get here:
telCh1:mem1ch1Jan2011ch1:
1st - determine a single bit
2nd - determine a full byte
3rd - determine the end of the single bit!
To see what ISL should do with this, just press **ENTER**.
Keystrokes needed for next screen
[ENTER]

第4章

- Keynotes enabled to get 'here':
[ENTER]
- As you can see, the **title** is made under the **level one heading** since that is the **level last relevance**.
- To find out more about lists in ISL, press **ENTER**.
- Keynotes needed for next screen:
[ENTER]

Keywords entered to get here:
[ENTER]
As you can see, the **Set** is made under the **level site heading** (this is the level last referenced).
To find out more about this in LSH, press **ENTER**.
Keywords needed for next screen
[ENTER]

四三三

- Revisions entered to get here:
[ENTER]
- There are two kinds of lists left: **unordered lists** and **ordered lists**.
The tags are similar to the simple list : **ul**, and **ol**, only the list type is different.
- Please enter and type the information presented in the Keystrokes Needed area from now on.
- Keystrokes needed for next screen:
[ENTER]

Variables entered to get here:
(ENTER)

There are two kinds of lists left, unordered lists and ordered lists.

The tags are similar to the simple list 'ul', and 'ol', only the list type is different.

Press enter and type the information presented in the Keystrokes Needed area from now on.

Keystrokes needed for next screen:
(ENTER)

四〇四

Diagram illustrating the architecture of a computer system. The main components shown are:

- 28: コマンド (Command)
- 26: イメージ (Image)
- 30: インストラクション (Instruction)
- 34: パーティナル (Terminal)
- 32: ドライバ (Driver)
- 33: モード・スイッチ (Mode Switch)
- 24: ハードディスク (Hard Disk)

Component 36 is a detailed view of a stack of disk drives:

1	36a	個人数据
2	36b	
3	36c	
4	36d	
5	36e	
6	36f	
7	36g	小磁
8	36h	

Annotations indicate connections between components 28, 26, and 30, and between 34, 32, 33, and 24. A dashed line encloses components 30 and 24. A bracket labeled "個人数据" (Personal Data) covers drives 1 through 6. A bracket labeled "小磁" (Small Magnet) covers drive 7. A bracket labeled "ユーティリティ/ドキュメンテーション" (Utility/Documentation) covers drive 8.

第5周

特開昭 62-214437 (33)

CMOS
sample of mistake here
.....1.....2.....ERROR IN INPUT, PLEASE RE-TRY6.....7.....USER

第6a図

CMOS
XEDIT IS DEMO SCRIPT A1
.....1.....2.....ERASE ERRORS, LEAVE CORRECTIONS6.....VM READ VM8

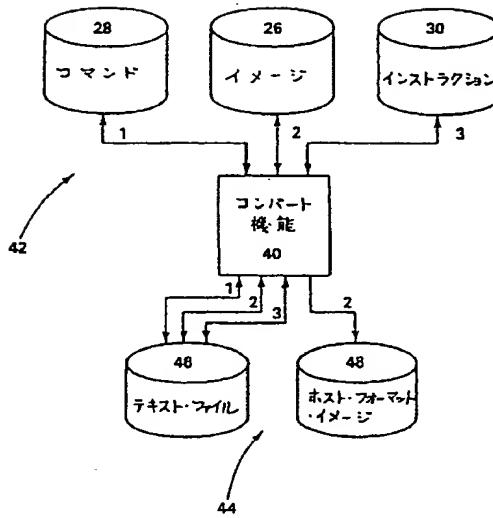
第6b図

CMOS
with bad demo script A1
.....1.....2.....DISPLAY IS THE PROPER ENTRY6.....7.....8
.....VM READ VM6

第6c図

CMOS
with bad demo script B1
.....1.....2.....PFS NOT CORRECT, USE: ENTER6.....7.....8
.....VM READ VM8

第6d図



第8図